

Entwicklung eines kompetenzbasierten Frameworks mit Aktivitätsschablonen und die Adaptierung des Child-Centered-Designs – eine empirische Studie zur frühen Technikbildung in Informatik

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der technischen Wissenschaften

eingereicht von

Mag.rer.nat. Kerstin Stöckelmayr

Matrikelnummer 09805224

an der

Fakultät für Informatik der Technische Universität Wien

Betreuung: Ao.Univ.Prof. Mag.rer.soc.oec. Dr.phil. Margit Pohl

Diese Dissertation haben begutachtet:

Ao.Univ.Prof. Dipl.-Ing. Dr.techn.
Gerald Futschek

Univ.Prof. i.R. Mag. Dr.
Wilfried Grossmann

Wien, 10.03.2019

Mag.rer.nat. Kerstin Stöckelmayr

Erklärung zur Verfassung der Arbeit

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Ort, Datum

Mag. Kerstin Stöckelmayr

Kurzfassung

Dem derzeit existierenden Fachkräftemangel in technischen Berufen, so auch der Informatik, soll durch Entwicklung geeigneter Frühbildungsmaßnahmen entgegengewirkt werden. Frühe Technikbildung bietet Kindern und später Jugendlichen die Möglichkeit, ihre Begeisterung und ihr Interesse für Technik zu entdecken und in ihren persönlichen Ausbildungsweg einfließen zu lassen.

Die vorliegende Forschungsarbeit beschäftigt sich, motiviert durch den voraussichtlich steigenden Mangel an hochqualifizierten Technikerinnen und Technikern, mit der Entwicklung und der empirischen Erforschung eines kompetenzbasierten Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ mit Aktivitätsschablonen für Kindergärten zur frühen Technikbildung in Informatik. Das für die Zielgruppe unter Beachtung didaktischer Aspekte wissenschaftlich fundierte Framework soll als Grundlage dienen, Kindern Technik, im Speziellen Informatik, und die Tätigkeiten einer Wissenschaftlerin bzw. eines Wissenschaftlers spielerisch näher zu bringen.

Die Grundlagen für das kompetenzbasierte Framework entstammen zum einen einer umfangreichen Literaturrecherche und zum anderen den Erfahrungen der Verfasserin der gegenständlichen Arbeit aus durchgeführten Vorstudien. Die Autorin entwickelt Aufgabenentwürfe, worauf aufbauend sie in zwei Kindergärten zahlreiche konkrete Aufgaben durchführt. Zur Ableitung der Aktivitätsschablonen, Finalisierung und empirischen Befundung des kompetenzbasierten Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ setzt sie die qualitative Inhaltsanalyse nach Mayring ein. Als Vorbereitung auf die Entwicklung und den Einsatz eines adaptierten Child-Centered-Design Ansatzes als Maßnahme zur frühen Technikbildung in Informatik führt die Verfasserin der gegenständlichen Arbeit Usability-Studien zu den im Framework eingesetzten Hard- und Software-Produkten durch.

Am Beispiel der Gestaltung und Erstellung eines Prototyps einer grafischen Benutzerschnittstelle für eine Mikrowelt, die für Kinder im Alter von 3 bis 6 Jahren geeignet ist, wird empirisch belegt, wie Child-Centered-Design, adaptiert und verfeinert um Cooperative-Inquiry und Bonded-Design, als weitere Maßnahme zur technischen Frühbildung in Informatik eingesetzt werden kann. Mit Hilfe von Kindern als Gestaltungspartnerinnen und -partner wird ein Prototyp einer Benutzerschnittstelle einer Mikrowelt entwickelt, die mittels Touchscreens in unterschiedlichen Einsatzszenarien als Lehr-Lern-Mittel genutzt werden kann. Mit dieser Maßnahme zur frühen Technikbildung soll Kindergartenkindern die Möglichkeit geboten werden, in unterschiedlichen Rollen (als Benutzerinnen, Benutzer, als Informantinnen, Informanten sowie als Gestaltungspartnerinnen und -partner oder auch als Testerinnen und Tester) einzelne Phasen der Software Entwicklung zu erleben und mitzugestalten. So soll den Kindergartenkindern vermittelt werden, dass Informatik weit mehr ist, als Programmieren oder einen PC zu bedienen.

Schlagwörter: Frühe Technikbildung in Informatik, Aktivitätsschablonen, Child-Centered-Design, Kindergarten, Prototyping

Abstract

The current skilled employees lack in technical professions, including computer science, should be counteracted by developing suitable early education actions. Early technology education offers children and later adolescents the opportunity to discover their enthusiasm and interest in technology and incorporate it into their personal learning path.

Motivated by the expected shortage of highly qualified technicians, the present research is concerned with the development and empirical research of a competence-based framework "Informatics for children aged 3 to 6 years" with activity templates for kindergartens for early technology education in computer science. The scientifically founded framework for the target group considering didactic aspects should serve as a basis to playfully introduce children to technology, in particular computer science, and the activities of a scientist.

The basis for the competence-based framework comes firstly from an extensive literature research and secondly from the experiences of the author of the present thesis from preliminary studies. The author develops task drafts, building up on them she performs numerous concrete tasks in two kindergartens. She uses qualitative content analysis according to Mayring, for deriving the activity templates, finalizing the framework and summing up the empirical findings of the competency-based framework "Informatics for children aged 3 to 6 years". In preparation for the development and use of an adapted child-centered design approach as an action of early technology education in computer science, the author of this thesis carries out usability studies on the hardware and software products used in the framework.

Using child-centered design, adapted and refined by cooperative inquiry and bonded design, as a further action for technical early education in computer science is empirically evidenced based on the example of designing and creating a prototype of a graphical user interface for a micro-world that is suitable for children aged 3 to 6 years. With the help of children as design partners, a prototype of a user interface of a micro-world is being developed, which can be operated by touchscreen and used as a resource for teaching and learning in various scenarios. With this action for early technology education, young children should be offered the opportunity to experience and participate in individual phases of software development in different roles (as users, informants, design partners and testers). Thus, the kindergarten children should be taught that computer science is much more than programming or using a PC.

Keywords: Early technology education, activity templates, child-centered design, kindergarten, prototyping

Hinweise

Alle innerhalb der Dissertation genannten und ggf. durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer. Allein aufgrund der bloßen Nennung ist nicht der Schluss zu ziehen, dass Markenzeichen nicht durch Rechte Dritter geschützt sind.

Die Nennung von Quellen erfolgt bei erstmaliger Verwendung unter Anführung aller Autorinnen und Autoren mit nachgestellter Jahreszahl der Veröffentlichung. Bei wiederholter Nennung der gleichen Quelle werden höchstens die ersten drei Autorinnen und Autoren, ggf. gefolgt von „et al.“ bei mehr als drei Autorinnen und Autoren, angeführt. Sollte die Kombination aus Autorinnen und Autoren sowie Jahreszahl für unterschiedliche Quellen mehrmals auftreten, werden ab der zweiten Quelle die Quellen durch einen Kleinbuchstaben, beginnend mit „b“, welcher der Jahreszahl folgt, differenziert.

Internetlinks sind als Fußnoten im Text, an der Stelle ihrer Verwendung, angeführt und mit dem Datum des letzten Zugriffs versehen.

Die Dokumentation der von der Autorin durchgeführten Studien in den Kindergärten erfolgte anonymisiert. Mit den Erziehungsberechtigten der an den Studien teilnehmenden Kinder vereinbarte die Autorin, dass die Daten anonymisiert, dem Kindergarten zugeordnet aber nicht rückverfolgbar auf die Person verwendet werden dürfen.

Inhaltsverzeichnis

Kurzfassung	I
Abstract	II
Hinweise	III
Inhaltsverzeichnis	V
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Ziele und Innovationen	1
1.3 Fragestellung und Forschungsdesign	2
1.3.1 Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“	3
1.3.2 Child-Centered-Design-Ansatz zur frühen Technikbildung	4
1.4 Erwartetes Resultat	5
1.4.1 Framework Informatik	5
1.4.2 Adaptierung des Child-Centered-Designs	6
1.5 Aufbau und Struktur der Arbeit	6
1.5.1 Theorieteil – State-of-the-Art	6
1.5.2 Praxisteil	7
1.6 Eigene Literatur und Randbedingungen	8
1.6.1 Eigene themenrelevante Publikationen	8
1.6.2 Randbedingungen	9
2 Grundlagen zur frühen Technikbildung	11
2.1 Lernen und Lerntheorien im Kontext der frühen Technikbildung	11
2.1.1 Fröbels Kindergarten	12
2.1.2 Konstruktivismus	12
2.1.3 Kognitive Entwicklung nach Piaget	13
2.1.4 Konstruktionismus nach Papert	14
2.1.5 ZPD – Zone of proximal development nach Vygotsky	15
2.2 Frühe Technikbildung im Kontext des lebenslangen Lernens, mit Schwerpunkt Österreich	16
2.3 IKT-Aus- und Weiterbildung von Kindergartenpädagoginnen und -pädagogen	19
2.3.1 Integration von IKT in Lehr-Lern-Szenarien	20
2.3.2 Herausforderungen	21
2.4 Lernen & Spiel – Lernen durch Spiel	22
2.5 Technik im Kontext der Geschlechter	24
2.5.1 Herangehensweisen von Mädchen und Jungen	24
2.5.2 Frauen in der Technikwelt	24
2.5.3 Frühzeitige Berührung mit Technik	26
2.5.4 Selbstbewusstsein und Rollenbilder	27
3 Informatikdidaktik und Computational Thinking	29
3.1 Didaktische Überlegungen zum Praxiseinsatz	29
3.1.1 Informatik – Was ist das?	31
3.1.2 Bildungswerte der Informatik	32
3.1.3 Basiskonzepte der Informatik	34
3.1.4 Algorithmisches Denken	36
3.2 Kompetenzentwicklung durch Einsatz von IKT und Robotern	36
3.2.1 Interaktion mit Objekten und Lernen	37
3.2.2 Problemlösen	39
3.2.3 Räumliche Orientierung	39
3.2.4 Soziale Kompetenz	40
3.2.5 Kompetenzentwicklung im Kindergartenalter	40
3.3 Higher-Order Thinking und Computational Thinking	41
4 3- bis 6-jährige Kinder lernen Programmieren	43
4.1 Einstieg in die Programmierung	43
4.2 Ansätze und Herausforderungen	45

4.3	Lernumgebungen zum Programmieren für Kinder	46
4.3.1	Software	47
4.3.2	Tangible Interfaces	50
4.3.3	Tangible Interfaces vs. Grafische Benutzeroberflächen	54
4.4	Educational Robotics	55
4.4.1	Robotics in Education	56
4.4.2	Exemplarische Bodenroboter	57
5	Child-Centered-Design	61
5.1	Einführung	61
5.1.1	Die Zielgruppe Kinder	62
5.1.2	Guidelines und Rahmenbedingungen	65
5.1.3	Evaluation von Eingabegeräten für Kinder	70
5.2	Usability Engineering mit (Kindergarten-)Kindern	72
5.2.1	Rollen	72
5.2.2	User-Centered-Design, die Basis für Child-Centered-Design	73
5.2.3	Weiterentwicklung des User-Centered-Designs für einen Einsatz mit Kindern als Gestaltungspartnerinnen und -partner	74
5.2.4	Card Sorting mit Kindergartenkindern	76
5.3	Usability Evaluation mit Kindern im Alter von 3 bis 6 Jahren	77
5.3.1	Thinking und Talk Aloud	77
5.3.2	Heuristische Evaluation	77
5.3.3	Prototyping mit Kindern	78
5.4	Weitere Forschungsergebnisse und Best-Practice-Beispiele	79
6	Forschungslücken und -bedarf	81
6.1	Research Gap: Einsatz von IKT im Bildungsbereich	81
6.2	Research Gap: Computational Thinking und Educational Robotics	84
6.3	Research Gap: Child-Centered-Design zur frühen Technikbildung in Informatik	90
6.4	Zwischenfazit: Frühe Technikbildung in Informatik	92
7	Forschungsmethoden	95
7.1	Die qualitative Feldforschung	96
7.1.1	Beobachtung im Kontext der qualitativen Feldforschung	97
7.1.2	Generalisierbarkeit und Validität qualitativer Methoden	98
7.2	Die qualitative Inhaltsanalyse nach Mayring	99
7.2.1	Von quantitativen Auswertungsmethoden zu qualitativen	99
7.2.2	Grundsätze der qualitativen Inhaltsanalyse	100
7.2.3	Drei Techniken zur Analyse: Zusammenfassung, Explikation und Strukturierung	100
8	Empirische Entwicklung eines Frameworks	105
8.1	Prozess zur Entwicklung des Frameworks	106
8.2	Phase 1: Leitthemen, Lernergebnisse und Kompetenzen im Framework	107
8.2.1	Leitthema: Early Childhood Technology	108
8.2.2	Leitthema: Computational Thinking	109
8.2.3	Leitthema: Child-Centered-Design	110
8.2.4	Zuordnung Lernergebnisse zu Kompetenzen	110
8.2.5	Entwicklung von Aufgabenentwürfen	111
8.2.6	Exemplarisch eingesetzte Hard- und Software	112
8.3	Phase 2: Extraktion von Aufgaben aus der durchgeführten empirischen Studie	113
8.3.1	Dokumentation	113
8.3.2	Begrifflichkeiten und Definitionen	114
8.3.3	Transkription und Kodierung, Inhaltsanalyse	115
8.3.4	Kategorien der Inhaltszusammenfassung	118
8.3.5	Extraktion von Metadaten der Aufgaben	119
8.4	Phase 3: Konsolidierung der Aufgaben in Inhaltsdimensionen und Aktivitätscluster	121

8.4.1	Inhaltsdimensionen.....	121
8.4.2	Aktivitätscluster	121
8.4.3	Aufgabenüberblick nach Inhaltsdimensionen und Aktivitätscluster	122
8.5	Phase 4: Entwicklung einer Tätigkeitslandkarte	125
8.6	Phase 5: Aufgaben-Matching zu Lernergebnissen.....	126
8.6.1	Überprüfung der Eignung der Aufgaben	126
8.6.2	Abschluss-Assessment	129
8.7	Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“	147
8.7.1	Framework Perspektive 1: Aktivitätsschablonen nach Inhaltsdimensionen.....	148
8.7.2	Framework Perspektive 2: Aktivitätsschablonen nach Leitthemen.....	162
8.7.3	Framework Perspektive 3: Aktivitätsschablonen nach Lernergebnissen.....	164
8.7.4	Framework Perspektive 4: Aktivitätsschablonen nach Tätigkeiten	165
8.8	Arbeiten mit dem Framework	166
8.8.1	Ableitung einer Aufgabe mit Perspektive 1 nach Inhaltsdimension	166
8.8.2	Ableitung einer Aufgabe mit Perspektive 2 nach Leitthema	168
8.8.3	Ableitung einer Aufgabe mit Perspektive 3 nach Lernergebnis	169
8.8.4	Ableitung einer Aufgabe mit Perspektive 4 nach Tätigkeit	170
9	Evaluierung von Hardware und Software	173
9.1	Evaluierung Hardware Usability	173
9.1.1	Funktionsweise Bee-Bot.....	173
9.1.2	Beschreibung des Tests	175
9.1.3	Eingesetzte Methoden und Vorgehensweise.....	176
9.1.4	Leitfaden zur zusammenfassenden Transkription.....	176
9.1.5	Kategoriensystem zur Kodierung.....	177
9.1.6	Ergebnisse	179
9.1.7	Zusammenfassende Interpretation der Ergebnisse und Schlussfolgerungen	181
9.2	Evaluierung Software Usability.....	185
9.2.1	Expertenreview Software Focus on Bee-Bot 2.....	185
9.2.2	Beschreibung des Tests	186
9.2.3	Fragestellung.....	188
9.2.4	Leitfaden zur zusammenfassenden Transkription.....	190
9.2.5	Kategoriensystem zur Kodierung.....	191
9.2.6	Ergebnisse	195
9.2.7	Zusammenfassende Interpretation der Ergebnisse und Schlussfolgerungen	204
10	Child-Centered-Design im Kindergarten	211
10.1	Ablauf des Child-Centered-Designs	212
10.2	Symbole gestalten	213
10.2.1	Gruppe 1	213
10.2.2	Gruppe 2	214
10.2.3	Gruppe 3	214
10.2.4	Eigene Symbole	214
10.2.5	Symbolauswahl für Card Sorting	214
10.3	Card Sorting - Symbole zuordnen	219
10.3.1	Setting und Durchführung.....	219
10.3.2	Endergebnis	221
10.4	Prototyp	226
10.4.1	Entwicklung des Prototyps – Version 1.....	226
10.4.2	Evaluation Prototyp Version 1	233
10.4.3	Iteration Prototyp - Version 2	239
10.4.4	Evaluation Prototyp Version 2	242

10.5	Beitrag des adaptierten Child-Centered-Designs zur frühen Technikbildung in Informatik	248
11	Konsolidierung der Ergebnisse und Fazit	251
11.1	Erzieltes Resultat.....	251
11.1.1	Framework Informatik.....	251
11.1.2	Adaptiertes Child-Centered-Design	251
11.1.3	Verbesserungspotentiale für Hard- und Softwareprodukte.....	252
11.2	Überprüfung der Hypothesen.....	253
11.2.1	H1.1	253
11.2.2	H1.2	254
11.2.3	H1.3	255
11.2.4	H1.4	255
11.2.5	H2.1	256
11.3	Beantwortung der Forschungsfragen	257
11.3.1	F1.1.....	259
11.3.2	F1.2.....	259
11.3.3	F1.3.....	260
11.3.4	F1.4.....	260
11.3.5	F2.1.....	261
11.3.6	F2.2.....	261
11.3.7	F2.3.....	262
11.3.8	F2.4.....	262
11.4	Fazit.....	262
11.5	Ausblick	264
11.6	Danksagung.....	266
	Literaturverzeichnis	267
	Abkürzungsverzeichnis.....	293
	Abbildungsverzeichnis.....	294
	Tabellenverzeichnis.....	295

Anhang

A1	Guidelines.....	A-1
A1.1	Fragen zur Evaluations- und Testgestaltung nach (Markopoulos et al., 2008)	A-1
A1.2	Handlungsempfehlungen zur Gestaltung von Usability-Tests nach (Hanna, Ridsen & Alexander, 1997)	A-2
A1.3	Zuordnung Evaluationsfragen zu Designphasen nach (Markopoulos et al., 2008)	A-4
A2	Konkrete durchgeführte Aufgaben der Studie.....	A-5
A2.1	Aufgaben Inhaltsdimension A	A-5
A2.2	Aufgaben Inhaltsdimension B	A-20
A2.3	Aufgaben Inhaltsdimension C	A-25
A2.4	Aufgaben Inhaltsdimension D	A-31
A2.5	Aufgaben Inhaltsdimension E	A-34
A3	Hinweise zur Kodierung Abschluss-Assessment.....	A-38
A3.1	AS01 Frage 1: Am besten gefallen	A-38
A3.2	AS02 Frage 2: Was ist der Bee-Bot	A-38
A3.3	AS03 Frage 3: Bee-Bot einschalten	A-38
A3.4	AS04 Frage 4: Funktion Bee-Bot	A-38
A3.5	AS05 Frage 5: Zwei Schritte vorwärts.....	A-39
A3.6	AS06 Frage 6: Bee-Bot losfahren lassen	A-39
A3.7	AS07 Frage 7: Komplexen Weg fahren.....	A-39
A3.8	AS08 Frage 8: Ohne Einzelschritte komplexen Weg.....	A-40
A3.9	AS09 Frage 9: Bee-Bot stoppen	A-40
A3.10	AS10 Frage 10: Ausschalten, Einschalten, GO.....	A-40

A3.11	AS11 Frage 11: Löschen	A-40
A3.12	AS12 Frage 12: Anzahl der Befehle, die gespeichert werden können	A-40
A3.13	AS13 Frage 13: Programmieren erklären.....	A-41
A3.14	AS14 Frage 14: Präferenzen	A-41
A3.15	AS15 Frage 15: Programme entwickeln.....	A-41
A3.16	AS16 Frage 16: Berufswunsch	A-41
A4	Hinweise zur Kodierung Hardware Usability	A-42
A4.1	HW01 Physikalische Eigenschaften des Bee-Bots.....	A-42
A4.2	HW02 Aktionen des Bee-Bots.....	A-42
A4.3	HW03 Bedienung des Bee-Bots.....	A-43
A4.4	HW04 Herangehensweisen und Eigenschaften des Kindes	A-43
A4.5	HW05 Interesse an Bee-Bot	A-43
A4.6	HW06 Analogien zur Software Focus on Bee-Bot 1 bzw. 2.....	A-43
A5	Hinweise zur Kodierung Software Usability	A-45
A5.1	SW01 Vorwissen	A-45
A5.2	SW02 Wahrnehmung.....	A-45
A5.3	SW03 Programmlogik / Benutzeroberfläche: Bedienung Software	A-46
A5.4	SW04 Sinngemäße Bedienung	A-47
A5.5	SW05 Orientierung	A-47
A5.6	SW06 Haptik.....	A-49
A5.7	SW07 Allgemeine Fehler	A-49
A5.8	SW08 Unterschiede	A-49
A5.9	SW09 Wissenstransfer	A-49
A5.10	SW10 Präferenzen.....	A-50
A5.11	SW11 Testsetting.....	A-50
A6	Symbole	A-52
A6.1	Symbole Gruppe 1	A-52
A6.2	Symbole Gruppe 2.....	A-53
A6.3	Symbole Gruppe 3.....	A-55
A6.4	Eigene Symbole.....	A-57
A7	Card Sorting	A-60
A8	Hinweise zur Kodierung Evaluation Prototyp 1	A-62
A9	Hinweise zur Kodierung Evaluation Prototyp 2	A-64

1 Einleitung

„However, we continue teaching our children about atoms and bits as two separate realms of experience. In the early schooling experiences, we teach them about polar bears and cacti, which are probably further from their everyday experience than smart faucets and cell phones.“ (Bers, 2008, S. 2)

1.1 Motivation und Problemstellung

Ausgangspunkte der Überlegungen bilden der laufende öffentliche österreichische und europaweite Diskurs hinsichtlich eines Mangels an hochqualifizierten Technikerinnen und Technikern am Arbeitsmarkt sowie die Betrachtung der Maßnahmen, die zur Technikbildung, insbesondere auch von Mädchen und Frauen in den jeweiligen Ländern gesetzt werden.

Als eine mögliche Herangehensweise kristallisiert sich in der Fachliteratur (siehe Kapitel 2) ein möglichst frühes „mit Technik in Berührung bringen“ heraus. Hintergedanke dabei ist, dass durch dieses frühe Kennenlernen von Technik und technischen Berufen und Geräten Berührungängste mit Technik minimiert bzw. idealerweise gar nicht erst aufgebaut werden. Im Mittelpunkt der Aktivitäten steht ein „Selbstverständnis von Technik“ – Technik und besonders auch technische Berufe sollen als etwas „ganz Normales“ und „Alltägliches“ sowie als positive Herausforderung (die bewältigbar ist) erlebt und verinnerlicht werden.

Zahlreiche Projektinitiativen (z.B. MINT, Kinderuni, FiT, Sparkling Science, ...) haben sich zum Ziel gesetzt, den technischen Nachwuchs auszubilden. Allerdings gibt es nach eingehender Recherche speziell in Österreich noch kein Konzept, das sich in bestehende Strukturen von Kindergärten nachhaltig einbinden lässt und so eine längerfristige frühe Technikbildung in Informatik ermöglicht. Unter früher Technikbildung in Informatik ist hier nicht die Bedienung eines Computers als „Werkzeug“ des Alltags zu verstehen, sondern die tatsächliche Auseinandersetzung mit informatischen Themen, wie zum Beispiel der Programmierung. Die beim Programmieren erworbenen Denkweisen und Problemlösungsfähigkeiten wirken sich auch auf den Zuwachs an Kompetenzen in anderen Lernbereichen positiv aus, wie der aktuellen Literatur, zusammengefasst aufbereitet in Kapitel 3.2 der gegenständlichen Arbeit, entnommen werden kann. Nach einer intensiven Literaturrecherche zeigt sich, dass noch kaum Forschungsergebnisse zur frühen technischen Bildung in Informatik in Kindergärten (3- bis 6-jährige Kinder) vorliegen. Kapitel 6 zeigt auf, dass sich die meisten empirischen Forschungsergebnisse mit Kindern ab 6 Jahren beschäftigen.

Die vorliegende Arbeit liefert neue empirische Erkenntnisse über den Einsatz von Bodenrobotern und Computern als Lehr-Lern-Hilfe in Kindergärten zur frühen Technikbildung in Informatik. Daraus resultiert als Innovation die Erstellung eines Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Dieses Framework beinhaltet Aktivitätsschablonen zur Entwicklung und Förderung von Informatikkompetenzen und zur Vorbereitung der 3- bis 6-jährigen Kinder auf die Teilnahme an einem Child-Centered-Design-Prozess. Die Adaptierung und Erweiterung eines per Literatur sehr eng definierten Child-Centered-Design-Ansatzes (siehe Kapitel 5.2.2) zu einer Maßnahme für die frühe Technikbildung in Informatik für 3- bis 6-jährige Kinder stellt die zweite Innovation dar. Exemplarisch wird mit Kindergartenkindern gemeinsam ein Prototyp einer geeigneten Benutzerschnittstelle für eine Mikrowelt zum Erlernen von Programmierung für Kindergartenkinder entwickelt. Durch die Teilnahme der 3- bis 6-jährigen Kinder an dem adaptierten Child-Centered-Design-Prozess kommen diese in unterschiedlichen Rollen mit der Entwicklung von Software in Berührung und lernen die Prozesse der Software Entwicklung kennen.

1.2 Ziele und Innovationen

Diese Forschungsarbeit liefert einen neuartigen Beitrag zur frühen Technikbildung in Informatik, der es Kindergärten nachhaltig erlaubt Kindern informatische Themen spielerisch näher zu bringen. Darüber hinaus präsentiert diese Arbeit empirische Erkenntnisse zu Child-

Centered-Design als Maßnahme zur frühen Technikbildung in Informatik. Folgende Innovationen sind Gegenstand der vorliegenden Forschungsarbeit:

1. Empirische Befunde und Forschungsergebnisse zur frühen Technikbildung in Informatik in Kindergärten,
2. Ein empirisch erforschtes, kompetenzbasiertes und verallgemeinertes Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zur nachhaltigen Vermittlung informatischer Themen,
3. Empirische Erkenntnisse über den Kompetenzerwerb von Kindergartenkindern in der Rolle der Lernenden beim Einsatz eines adaptierten Child-Centered-Design-Prozesses als Maßnahme zur frühen Technikbildung in Informatik für 3- bis 6-jährige Kinder,
4. Empirische Befundung des Einsatzes von Child-Centered-Design als Maßnahme zur frühen Technikbildung in Informatik, bei der Kindergartenkinder in den Rollen Benutzerinnen, Benutzer, Testerinnen, Tester, Informantinnen, Informanten sowie Gestaltungspartnerinnen und -partner (nach Nasset & Large, 2004) mit der Software Entwicklung, einem der kreativsten Bereiche der Informatik, in Berührung kommen.

Daraus abgeleitet, lassen sich für die gegenständliche Forschungsarbeit folgende Ziele formulieren:

1. Empirische Erforschung eines allgemeinen Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ im Kindergarten zur frühen Technikbildung in Informatik unter Berücksichtigung des voraussichtlich steigenden Mangels an hochqualifizierten Technikerinnen und Technikern,
2. Sammlung von empirischen Erkenntnissen über den Einsatz eines adaptierten Child-Centered-Design-Ansatzes als Maßnahme zur frühen Technikbildung in der Informatik am Beispiel der Entwicklung und Evaluierung eines Prototyps einer grafischen Benutzerschnittstelle für eine Mikrowelt für Kinder im Alter von 3 bis 6 Jahren.

1.3 Fragestellung und Forschungsdesign

Die Autorin leitet ihre Fragestellungen zum Thema frühe Technikbildung in Informatik für Kindergartenkinder zum einen aus der im Vorfeld durchgeführten Vorstudie (siehe Kapitel 1.4) und zum anderen theoriegeleitet auf Basis einer umfangreichen Literaturanalyse (Kapitel 2, 3, 4 und 5) ab.

Der Praxisteil der Forschungsarbeit beschäftigt sich zu Beginn mit der wissenschaftlich fundierten Entwicklung und empirischen Analyse eines Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Das Framework dient als Antwort auf die Forschungsfrage, ob und wie Kindergartenkindern die Grundprinzipien der Programmierung vermittelt werden können. Anschließend widmet sich die Verfasserin der gegenständlichen Arbeit dem Child-Centered-Design als Methode zur frühen Technikbildung in Informatik in Österreich. Hierzu adaptiert und verfeinert sie als Innovation den in der Literatur definierten Child-Centered-Design-Ansatz um Cooperative-Inquiry und Bonded-Design (siehe Kapitel 5.2.3.3). Anschließend legt sie die empirischen Ergebnisse, die bei der Durchführung der Child-Centered-Design-Aktivitäten als Maßnahme zur frühen Technikbildung in Informatik am Beispiel der Entwicklung und Evaluation eines Prototyps einer grafischen Benutzerschnittstelle einer Mikrowelt am PC für Kindergartenkinder im Alter von 3 bis 6 Jahren gewonnen wurden, dar. Resümiert wird diese Arbeit mit ersten empirischen Erkenntnissen zur Kompetenzförderung teilnehmender Kinder – sie sind nicht nur Teilnehmerinnen und Teilnehmer am Prozess, sondern auch Lernende.

1.3.1 Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Zum Einstieg in den Praxisteil der vorliegenden Forschungsarbeit entwickelt die Autorin ein Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Für das Framework sammelt die Verfasserin der gegenständlichen Arbeit empirische Befunde in zwei Kindergärten und bildet somit die Grundlage für die weiteren Forschungstätigkeiten.

1.3.1.1 Forschungsfragen

Zur strukturierten und zielgerichteten Bearbeitung des Forschungsthemas, stellte die Autorin folgende aus der Vorstudie (siehe Kapitel 1.4) und der Theorie (siehe Kapitel 2 bis 4) abgeleitete Forschungsfragen auf:

F1.1: Wie können mögliche didaktische Prinzipien und Aktivitätsschablonen aussehen, im Zuge derer Kinder die Programmierung einfacher grundlegender Befehlsfolgen erlernen können?

F1.2: Gelingt 3- bis 6-jährigen Kindern der Lerntransfer der mit einem Bodenroboter erlernten Programmierbefehlsabfolgen auf die Programmierumgebung / Mikrowelt des virtuellen Bodenroboters (=Simulation)?

F1.3: Können Kinder im Alter von 3 bis 6 Jahren Usability-Schwächen von zielgruppengerechten Bodenrobotern und Programmierumgebungen / Mikrowelten am PC identifizieren?

F1.4: Sind geschlechtsspezifische Unterschiede hinsichtlich der Herangehensweise, des generellen Interesses und der Begeisterung beobachtbar?

1.3.1.2 Hypothesen

Aus den Forschungsfragen bildete die Verfasserin der gegenständlichen Arbeit folgende Hypothesen:

H1.1: Wenn 3- bis 6-jährige Kinder altersgerecht in der Programmierung einfacher, grundlegender Befehlsabfolgen geschult werden, dann können sie Problemstellungen desselben Schwierigkeitsgrades nach Teilnahme an Lehr-Lern-Einheiten, die aus dem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ abgeleitet werden können, selbständig lösen.

H1.2: Wenn Kinder im Alter von 3 bis 6 Jahren in der Programmierung eines Bodenroboters geschult werden, dann können sie selbständig ohne zusätzliche Einführung eine eingesetzte Programmierumgebung / Mikrowelt des virtuellen Bodenroboters am PC (=Simulation) bedienen, um einfache, grundlegende Befehlsabfolgen auszuführen.

H1.3: Wenn 3- bis 6-jährige Kinder in der Bedienung der zum Programmieren eingesetzten zielgruppengerechten Hard- und Softwareprodukte geschult werden, dann können sie Usability-Schwächen derselben, die mit einem Expertinnen- und Experten-Review korrespondieren, identifizieren.

H1.4: Wenn Kinder im Alter von 3 bis 6 Jahren, unter der Voraussetzung, dass in beiden Fällen das gleiche Lehr-Lern-Einheiten-Programm zum Einsatz kommt, bei dem nur die Reihenfolge der Aktivitäten verändert wird, Programmieren lernen, in dem sie zuerst einen Hardware Bodenroboter und anschließend eine korrespondierende Softwaresimulation (Programmierumgebung / Mikrowelt) nutzen, dann erzielen diese Kinder bessere Ergebnisse bei einem Assessment als die Kinder, die zum Programmieren lernen zuerst die Softwaresimulation (Programmierumgebung / Mikrowelt) und anschließend den korrespondierenden Hardware Bodenroboter nutzen.

1.3.1.3 Methoden

Zur Beantwortung der Forschungsfragen und zur Prüfung der Hypothesen wird eine explorative experimentelle Felduntersuchung zur empirischen Befundung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“, ergänzt um

- Literaturstudium (Ad F1.1 und F1.4),
- Usability-Tests: Leitfadengestützte Benutzerinnen- und Benutzertests mit Screen- und Videorecording sowie freie systematische Beobachtungen (Ad F1.3, H1.3),
- Qualitative Inhaltsanalyse nach Mayring in Kombination mit Leistungstests¹ (Ad H1.1),
- Teilnehmende-offene halbstandardisierte systematische Beobachtungen (Ad F1.1, F1.2, F1.3, F1.4, H1.2, H1.3 und H1.4) und
- Systematische apparative Beobachtungen (Ad H1.2, H1.3 und H1.4)

eingesetzt.

1.3.2 Child-Centered-Design-Ansatz zur frühen Technikbildung

In der vorliegenden Forschungsarbeit gilt es weiter zu klären, ob ein um Bonded-Design und Cooperative Inquiry erweiterter Child-Centered-Design-Ansatz zur frühen Technikbildung in Informatik für Kindergartenkinder in Österreich eingesetzt werden kann. Dies wird am Beispiel einer Entwicklung eines Prototyps einer grafischen Benutzerschnittstelle einer Mikrowelt, die auf einer Mini-Language basiert und für die Zielgruppe Kindergartenkinder geeignet ist, untersucht.

1.3.2.1 Forschungsfragen

Zur Bearbeitung des Themas orientiert sich die Verfasserin der gegenständlichen Arbeit an folgenden aus der Vorstudie (siehe Kapitel 1.4) und der Theorie (siehe Kapitel 5) hergeleiteten Forschungsfragen:

F2.1: Welche Möglichkeiten bestehen für einen Child-Centered-Design-Ansatz in einem österreichischen Kindergarten und welche Materialien können hierzu eingesetzt werden?

F2.2: Kann mit Hilfe eines Child-Centered-Design-Prozesses in Kombination mit Prototyping mit 3- bis 6-jährigen Kindern eine prototypische Benutzerschnittstelle einer Mikrowelt entwickelt werden, die auf einer Mini-Language basiert und vom Bee-Bot inspiriert ist?

F2.3: Wie kann eine unter Einsatz eines Child-Centered-Design-Prozesses entwickelte benutzerfreundliche grafische Benutzerschnittstelle einer zielgruppengerechten Mikrowelt am PC gestaltet sein?

F2.4: Wie muss Child-Centered-Design weiterentwickelt werden, um als Methode zur frühen Technikbildung in Informatik für Kinder im Alter von 3 bis 6 Jahren eingesetzt werden zu können?

1.3.2.2 Hypothese

H2.1: Wenn Child-Centered-Design mit 3- bis 6-jährigen Kindern als Methode zur Entwicklung eines Prototyps einer grafischen Benutzerschnittstelle für eine zielgruppengerechte Mikrowelt eingesetzt wird und Kinder in den gesamten Design-Prozess involviert werden, dann entsteht ein Prototyp, der als Basis für die Entwicklung einer zielgruppengerechten Mikrowelt dienen kann und dessen Bedienelemente von 3- bis 6-jährigen Kindern, die nicht am Design-Prozess teilgenommen haben, selbständig und ohne Einführung genutzt werden können.

¹ Als Leistungstests, eine quantitative Methode, bezeichnet man Aufgaben, die objektiv „richtig“ oder „falsch“ beurteilt werden können „[...] d.h. wenn ein Beurteilungsmaßstab vorliegt. [...] Zur Gruppe der Leistungstests zählen Intelligenztests, Entwicklungstests, Schultests, allgemeine Leistungstests und spezielle Funktions- und Eignungstests.“ (Bortz & Döring, 2006, S. 190)

1.3.2.3 Methoden

Zur Beantwortung der Forschungsfragen und Prüfung der Hypothese zum Child-Centered-Design-Ansatz zur frühen Technikbildung in Informatik der gegenständlichen Forschungsarbeit bedient sich die Autorin einer explorativen experimentellen Felduntersuchung, die mittels

- Literaturanalyse (Ad F2.4 und H2.1),
- Methoden des CCDs (Child-Centered-Designs) (Ad F2.2 und H2.1),
- Usability-Evaluierungen mit Talk Aloud (Ad F2.2, F2.3 und H2.1),
- Teilnehmenden-offenen halbstandardisierten systematischen Beobachtungen (Ad F2.1) und
- apparativen Beobachtungen (Ad F2.1)

ergänzt wird.

1.4 Erwartetes Resultat

Die Vision der vorliegenden Forschungsarbeit ist die Bereitstellung eines wissenschaftlich fundierten, empirisch belegten Technikbildungskonzepts für Kindergartenkinder im Bereich der Informatik unter Einsatz von zielgruppengerechten Bodenrobotern und Programmierumgebungen sowie von Child-Centered-Design-Methoden.

Als Vorarbeit und Vorstudie fand im Herbst 2010 mit 5- bis 6-jährigen Kindergartenkindern (Vorschulkindern) ein zehn Einheiten umfassender Kurs in einem Kindergarten in Wien (Kindergarten A) statt. Dieser bot eine erste Einführung in die Programmierung mit Hilfe von Bodenrobotern wie beispielsweise dem Bee-Bot der Firma TTS Group. Darüber hinaus konnten mit Hilfe der Vorarbeiten Erkenntnisse über die einzusetzenden Forschungsmethoden und die Forschungsfragestellungen für die gegenständliche Forschungsarbeit gewonnen werden.

Ein Erfahrungsbericht hierzu und die Rahmenbedingungen sind zu finden in

Stöckelmayr, K., Tesar, M., & Hofmann, A. (2011). Kindergarten Children Programming Robots: A First Attempt. In R. Stelzer & K. Jafarmadar (Eds.), Proceedings of the 2nd International Conference on Robotics in Education, RiE 2011 (pp. 185 – 192). Vienna, Austria; September, 15-16, 2011, ISBN: 978-3-200-02273-7. Retrieved from http://www.robotchallenge.org/fileadmin/user_upload/_temp_/RiE/Proceedings/51.pdf

Die eben genannte Publikation wird in der vorliegenden Forschungsarbeit als Vorstudie referenziert bzw. genannt.

1.4.1 Framework Informatik

Ziel ist es, den Kindern spielerisch Technik näher zu bringen, das Berufsbild der Technikerin, des Technikers und der Forscherin, des Forschers vorzustellen und Berührungsängste mit Technologien abzubauen oder erst gar nicht entstehen zu lassen. Darüber hinaus soll aufgezeigt werden, dass Kindergartenkindern Informatikwissen vermittelt werden kann und die Kinder dieses erwerben können. Als Basis dient dafür ein von der Autorin entwickeltes, wissenschaftlich und theoretisch fundiertes, empirisch belegtes sowie kompetenzbasiertes Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“.

Begleitend erarbeitet die Autorin Hilfestellungen für zukünftige Anwenderinnen und Anwender bezüglich der Rahmenbedingungen von frühen Technikbildungsprogrammen. Zudem legt sie spezielles Augenmerk auf die Usability der für die Programmierereinführung eingesetzten Hard- und Software.

Mit diesem empirisch entwickelten und evaluierten Framework, das auf aktuellen Forschungserkenntnissen beruht und in der Praxis eingesetzt werden kann, wird ein wesentlicher Beitrag zur frühen Technikbildung in Informatik für Kinder im Alter von 3 bis 6

Jahren geleistet. Darüber hinaus werden den Kindern erste Ansätze einer wissenschaftlichen Vorgehensweise vorgestellt.

1.4.2 Adaptierung des Child-Centered-Designs

Mithilfe eines adaptierten Child-Centered-Design-Ansatzes, in Kombination mit Prototyping, mit Kindern im Alter von 3 bis 6 Jahren, wird exemplarisch eine benutzerfreundliche grafische Benutzerschnittstelle am PC für die Zielgruppe entwickelt. Hierbei werden nicht nur empirische Befunde ob des Einsatzes eines Child-Centered-Design-Prozesses mit Kindergartenkindern gesammelt, sondern auch ob der Möglichkeit, Child-Centered-Design als Maßnahme zur frühen Technikbildung in Informatik für Kindergartenkinder einsetzen zu können. Damit soll den Kindern im Kindergartenalter die Möglichkeit geboten werden, in unterschiedlichen Rollen einzelne Phasen der Software Entwicklung erleben und mitgestalten zu können. Ziel ist es den Kindergartenkindern zu vermitteln, dass Informatik weit mehr ist, als Programmieren oder einen PC zu bedienen.

1.5 Aufbau und Struktur der Arbeit

Die Arbeit umfasst einen Theorieteil und einen Praxisteil. Zu Beginn legt die Autorin den aktuellen Stand der Forschung, sowie die Erkenntnisse aus dem Literaturstudium als Grundlage zur späteren Beantwortung der Forschungsfragen dar. Es folgt die praktische Durchführung der Forschung mit den genannten Forschungsmethoden.

Zum Abschluss präsentiert die Verfasserin der gegenständlichen Arbeit die Ergebnisse der Forschungsarbeit, prüft die Forschungshypothesen und beantwortet die Forschungsfragen mit Hilfe der Erkenntnisse aus dem Literaturstudium und der praktischen Forschungsarbeit.

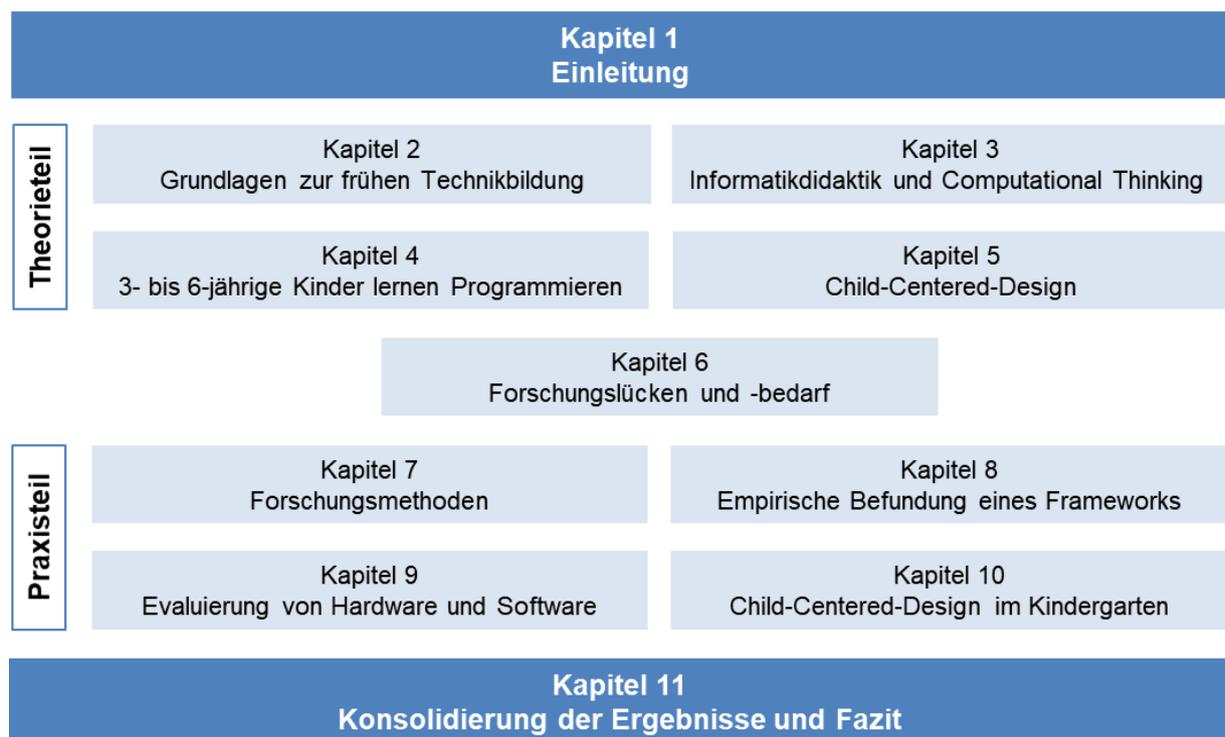


Abbildung 1: Aufbau der Arbeit

1.5.1 Theorieteil – State-of-the-Art

Die gegenständliche Arbeit beginnt mit dem Theorieteil. Dieser schafft mit einer Literaturrecherche sowie -aufbereitung einen Überblick über den aktuellen Stand der Forschung im Bereich der Technikbildung (Kapitel 2). Dieser Abschnitt dient der Erörterung der Grundlagen für eine Technikbildung im Kontext des lebenslangen Lernens. Die

Darstellung der Bedeutung des Spiels für das Lernen und des aktuellen Stands der Literatur zum Thema Gender und Technik runden dieses Kapitel ab.

Im nächsten Schritt folgt eine Aufbereitung der Grundlagen zum Programmieren für Kinder (Kapitel 3). Hierzu gehören eine Betrachtung der Didaktik der Informatik sowie eine Vorstellung von Basiskonzepten der Informatik und eine Darlegung der Kompetenzentwicklung, die durch Programmieren gefördert werden kann. Die Erörterung des Begriffs „Programmieren“ im Kontext von Kindern ist ebenso Bestandteil dieses Kapitels wie die Diskussion des „Higher-Order Thinkings“, des „Computational Thinkings“ und des algorithmischen Denkens.

Kapitel 4 beschäftigt sich mit dem Erlernen der Programmierung für Kinder im Kindergartenalter (3 bis 6 Jahre). Die Vorstellung von Produkten und Entwicklungen im Bereich der Software-Programmierumgebungen, der Tangible Interfaces und Robotik wird von didaktischen Überlegungen zum Einsatz von Programmieren im Unterricht begleitet

In Kapitel 5 findet eine Einführung in das Thema Child-Centered-Design statt. Wobei hier nicht auf die Grundlagen der Usability eingegangen wird, sondern auf die Besonderheiten im Kontext der Zielgruppe Kindergartenkinder. Neben der Diskussion von Eingabegeräten für Kinder, steht auch die Gestaltung von Usability Engineering Prozessen, von Usability Evaluationen sowie die Zusammenarbeit mit Kindern im Kindergartenalter im Fokus. Eine Übersicht über die notwendigen Methoden sowie die möglichen Implikationen auf den Child-Centered-Design-Prozess an sich schließen das Kapitel ab.

Der Theorieteil resümiert in Kapitel 6 mit einem Aufzeigen der aktuellen Forschungslücke zur Begründung der gegenständlichen Arbeit zu den Themen Programmieren für Kinder, Robotics in Education, Child-Centered-Design als Maßnahme zur frühen Technikbildung sowie mit einem Zwischenfazit.

1.5.2 Praxisteil

Der Einstieg in den Praxisteil erfolgt über die Vorstellung der zum Einsatz kommenden Forschungsmethoden (Kapitel 7). Neben der Definition einiger für diese Arbeit relevanter Begriffe aus der Forschungsmethodik, beinhaltet dieses Kapitel die Vorstellung der qualitativen Inhaltsanalyse nach Mayring, die als Basis-Forschungsmethode der gegenständlichen Arbeit dient.

In Kapitel 8 führt die Autorin die empirische Befundung eines Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ durch. Neben einer Detailbeschreibung der konkreten Aufgaben, die durchgeführt wurden und der verallgemeinerten generalisierten Aktivitätsschablonen des Frameworks beinhaltet das Kapitel den Prozess der Erstellung des Frameworks, die Darlegung der gesammelten empirischen Erkenntnisse sowie die Ergebnisse über den Kompetenzerwerb der Kinder, die Aufgaben, die der Entwicklung der Aktivitätsschablonen des Frameworks zugrunde liegen, durchführten.

Anschließend erfolgt eine Usability-Evaluierung der eingesetzten Hard- und Software. Hierbei agieren Kindergartenkinder in der Rolle der Testerin oder des Testers. Diese Evaluierungen sind in Kapitel 9 zu finden. Die Verfasserin der gegenständlichen Arbeit legt die eingesetzte Test- und Auswertungsmethodik und die jeweiligen Ergebnisse der Studien im Detail dar. Darüber hinaus wird der Einsatz der Usability-Evaluierungen als Methode zur frühen Technikbildung in Informatik erörtert.

In Kapitel 10 behandelt die Autorin den adaptierten Child-Centered-Design-Prozess mit Kindergartenkindern. Hierzu zeigt sie nicht nur die Vorgehensweise, sondern auch Erkenntnisse über den Einsatz auf und präsentiert das Endergebnis dieses kooperativen Prozesses mit Kindern, der exemplarisch zur prototypischen Entwicklung einer Usability-optimierten grafischen Benutzerschnittstelle für eine zielgruppengerechte Mikrowelt zur Simulation der Bedienung eines Bodenroboters, durchgeführt wird. In diesem Prozess wirken die Kinder in unterschiedlichen Rollen mit: zum einen als Benutzerinnen, Benutzer, zum anderen als Testerinnen, Tester, Informantinnen, Informanten sowie Gestaltungs-

partnerinnen und -partner. Abgerundet wird das Kapitel mit einer Darlegung, welche Kompetenzen der Kinder im Kindergartenalter als Lernende durch eine Teilnahme an dem adaptierten Child-Centered-Design-Prozess in allen Rollen gefördert werden können.

Die konsolidierte Beantwortung der Hypothesen und Forschungsfragen aus Kapitel 1.3 erfolgt in Kapitel 11 und ein zusammenfassendes Fazit rundet die gegenständliche Forschungsarbeit ab.

1.6 Eigene Literatur und Randbedingungen

Im folgenden Abschnitt sind die eigenen themenrelevanten Publikationen der Verfasserin der gegenständlichen Arbeit (Kapitel 1.6.1) sowie die berücksichtigten Randbedingungen (Kapitel 1.6.2) für diese Forschungsarbeit angeführt.

1.6.1 Eigene themenrelevante Publikationen

- Pucher, R., Salzbrunn, B., Stöckelmayr, K., & Tesar, M. (2011). Mobile Learning Apps and Educational Web-Platforms for Organizational Aspects of Teaching. In *Proceedings CD EDULearn11*. Barcelona, Spain, July 4-6, 2011, ISBN: 978-84-615-0441-1.
- Salzbrunn, B., Tesar, M., Stöckelmayr, K., & Pucher, R. (2011). Early Software Prototyping for Usability Tests using low cost tools for teaching purposes. In *Proceedings of the International Technology, Education and Development Conference (INTED) 2011*. Valencia, Spain; March 7-9, 2011, ISBN: 978-84-614-7423-3.
- Stöckelmayr, K., Tesar, M., & Hofmann, A. (2011). Kindergarten Children Programming Robots: A First Attempt. In R. Stelzer & K. Jafarmadar (Eds.), *Proceedings of the 2nd International Conference on Robotics in Education, RiE 2011* (pp. 185 – 192). Vienna, Austria; September, 15-16, 2011, ISBN: 978-3-200-02273-7. Retrieved from http://www.robotchallenge.org/fileadmin/user_upload/_temp_/RiE/Proceedings/51.pdf
- Stöckelmayr, K., Tesar, M., Hofmann, A., Pucher, R., & Schmöllebeck, F. (2011). New Role Models for IT-Students to Improve their Motivation to Learn. In T. Bastiaens & M. Ebner (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011* (pp. 2318–2323). Lisbon, Portugal; June 27 - July 1, 2011: AACE.
- Tesar, M., Pucher, R., Stöckelmayr, K., Metscher, J., Vohle, F., & Ebner, M. (2011). Interaktive, multimediale Materialien – Gestaltung von Materialien zum Lernen und Lehren. In M. Ebner & S. Schön (Eds.), *L3T – Lehrbuch für Lernen und Lehren mit Technologien* (1. Auflage.). Retrieved from <http://l3t.eu/homepage/das-buch/ebook/kapitel/lesen/o/id/38>
- Tesar, M., Stöckelmayr, K., Pucher, R., Ebner, M., Metscher, J., & Vohle, F. (2013). Multimediale und interaktive Materialien – Gestaltung von Materialien zum Lernen und Lehren. In M. Ebner & S. Schön (Eds.), *L3T – Lehrbuch für Lernen und Lehren mit Technologien* (2. Auflage.). Retrieved from <http://l3t.eu/homepage/das-buch/ebook-2013/kapitel/lesen/o/id/98>
- Tesar, M., Stöckelmayr, K., Sieber, S., & Pucher, R. (2011). Agilität als Chance zum Qualitätsmanagement in modernen Lehr-Lern-Szenarien. In T. Köhler & J. Neumann (Eds.), *Wissensgemeinschaften, Digitale Medien – Öffnung und Offenheit in Forschung und Lehre; Medien in der Wissenschaft, Band 60* (pp. 157–167). Waxmann Verlag, ISBN 978-3-8309-2545-3.
- Zorn, I., Trappe, C., Stöckelmayr, K., Kohn, T., & Derndorfer, C. (2013). Interessen und Kompetenzen fördern – Programmieren und kreatives Konstruieren. In M. Ebner & S. Schön (Eds.), *L3T – Lehrbuch für Lernen und Lehren mit Technologien* (2. Auflage.). Retrieved from <http://l3t.eu/homepage/das-buch/ebook-2013/kapitel/lesen/o/id/142>

1.6.2 Randbedingungen

In der gegenständlichen Arbeit betrachtet die Autorin die Zielgruppe der jungen Kinder, die in der Literatur mit einem Alter von ca. 4 bis 7 Jahren definiert wird. Speziell im Fokus der Verfasserin der gegenständlichen Arbeit liegen Kinder, die im Kindergarten bzw. in der Vorschule aktiv sind oder von ihrem Alter her in der Lage wären einen Kindergarten oder eine Vorschule zu besuchen. Daher werden die Altersgrenzen für die Zielgruppe von der Autorin auf ca. 3 bis 6 Jahre eingegrenzt. Als ältere Kinder werden in der Literatur Kinder ab 8 Jahren betrachtet, in der gegenständlichen Arbeit ab 7 Jahren. Näheres zu dieser Abgrenzung ist in Kapitel 5.1.1 zu finden.

2 Grundlagen zur frühen Technikbildung

In diesem Kapitel behandelt die Autorin die Voraussetzungen im Bildungssystem zur Einführung von IKT und Programmieren für 3- bis 6-jährige Kinder. Zu Beginn beleuchtet sie die Bedeutung der Kindergärten und der vorschulischen Bildung. Darauf folgen die Lerntheorien und Voraussetzungen zum Lernen, die im Kontext der frühen Technikbildung als wesentlich erachtet werden. Dazu zählen Fröbels Kindergarten, der Konstruktivismus, die Entwicklung des Kindes nach Piaget, der Konstruktivismus nach Papert sowie die „Zone of proximal development“ nach Vygotsky. (Kapitel 2.1)

Die frühe Technikbildung ist im Kontext des lebenslangen Lernens zu betrachten, welches in jüngster Zeit durch den Europäischen Qualifikationsrahmen deutlich an Bedeutung gewonnen hat. Hier gilt es für die Verfasserin der gegenständlichen Arbeit die zu erlangenden Kompetenzen in Hinblick auf die Digitalisierung unserer Zeit in Korrelation zu den Bildungsrahmenplänen elementarer Bildungseinrichtungen in Österreich, Deutschland und der Schweiz sowie der EU zu erörtern. (Kapitel 2.2)

Um neue Lehrinhalte vermitteln zu können, bedarf es entsprechend ausgebildeter Lehrkräfte. Hierzu wird die Aus- und Weiterbildung von Kindergartenpädagoginnen und -pädagogen sowie Lehrerinnen und Lehrern aufgegriffen und im Kontext der IKT beleuchtet. (Kapitel 2.3)

Die Integration des Spiels in den Unterricht wird in der Literatur als wesentlich für die Zielgruppe zur Vermittlung von IKT erachtet. Ebenso behandelt die Autorin die Thematik Gender und Technik, wobei sie herausarbeitet, ob Unterschiede und wenn ja, welche zwischen den Geschlechtern in Bezug auf Technik und IKT bei jungen Kindern existieren. (Kapitel 2.4 und 2.5)

In der Literatur und in der Praxis besteht der Konsens, dass die frühe Kindheit nicht nur eine sensible Phase des Lebens ist, sondern auch durch die Entwicklung kritischer kognitiver, sozialer, emotionaler und körperlicher Fähigkeiten gekennzeichnet ist. (Hustedt & Barnett, 2010, S. 110) Nicht zuletzt deswegen haben in den letzten Jahren die Bemühungen um vorschulische Bildung und Kindergärten, im Speziellen für 3- bis 6-Jährige zugenommen. (Hustedt & Barnett, 2010, S. 118)

Die Vielfältigkeit und Qualität früher Erfahrungen bestimmen die Dimension der neuronalen Grundlagen für zukünftiges Lernen, soziales Verhalten und Beziehungen. Auch motivieren frühe Lernerfahrungen Kinder zu ihren späteren schulischen Lernleistungen und langfristigen schulischen Erfolgen. Frühe Interaktionen und Beziehungen mit Erwachsenen und gleichaltrigen Peers legen den Grundstein für das zukünftige Sozialverhalten, die emotionale Sicherheit und das Selbstwertgefühl. (Thorpe, Cloney & Tayler, 2010, S. 144)

„Although young children are growing up in an increasingly digital environment, school curriculum does not always focus on exploring the digital world until later elementary years. Only a small number of countries and regions (such as the United Kingdom) have established clear policies and frameworks for introducing technology to young children [...].“ (Sullivan & Bers, 2015, S. 2)

Die von den Staaten und Institutionen angebotenen Initiativen in der Frühbildung fallen sehr divers aus und zeigen, dass sich kulturelle und soziale sowie auch infrastrukturelle Einflüsse direkt auf die Qualität und Effektivität dieser Initiativen auswirken. (Janus & Brinkman, 2010, S. 29) Evaluierungen und Diskussionen sind daher immer im jeweiligen kulturellen und sozialen Kontext zu sehen und Vergleiche mit anderen Angeboten differenter Kulturen stets mit Vorsicht zu genießen.

2.1 Lernen und Lerntheorien im Kontext der frühen Technikbildung

Lehren und Lernen im Kontext der frühen Technikbildung stellt zum einen eine Herausforderung dar, zum anderen gibt es zahlreiche Grundlagen, die eine solide Basis bilden, auf der aufgebaut werden kann. Wesentliche Bausteine für die Integration von IKT in Kindergärten und Schulen bringen Fröbels Kindergarten, der Konstruktivismus, die Theorie

der kognitiven Entwicklung nach Piaget, der Konstruktivismus nach Papert und Vygotskys „Zone of proximal development“ mit ein.

2.1.1 Fröbels Kindergarten

Die Kindergärten haben sich seit ihrem Aufkommen vor knapp 200 Jahren sehr verändert. Waren früher Geschichten erzählen, Schlösser bauen, Bilder zeichnen und Teilen lernen an der Tagesordnung, so wird der Kindergarten heute immer mehr zu einer Vorschule, in der Kinder ähnlich wie in der Schule lernen. (Resnick, 2007, S. 1)

„In my mind, exactly the opposite is needed: Instead of making kindergarten like the rest of school, we need to make the rest of school (indeed, the rest of life) more like kindergarten.“ (Resnick, 2007, S. 2)

Resnick (2007) bricht hier eine Lanze für informelles Lernen, das aus sechs spiralförmig angeordneten Lernaspekten besteht. Kinder stellen sich vor, was sie tun wollen. Darauf aufbauend kreieren sie ein Projekt, spielen mit ihren Ideen, teilen ihre Ideen und Kreationen mit anderen, reflektieren darüber und entwickeln neue Ideen und entwerfen wiederum Neues. Diesen Lernansatz, zum einen selbstentdeckend, zum anderen sehr kreativ und sozial, möchte Resnick (2007) auf alle Altersstufen übertragen und so jedem die Chance bieten, eine kreative Denkerin, ein kreativer Denker zu werden. (Resnick, 2007, S. 2) Das Erstellen von Neuem, das Kreieren, ist das zentrale Element eines kreativen Prozesses. Um Kinder dabei zu unterstützen, bedarf es geeigneter Hilfsmittel. (Resnick, 2007, S. 3)

Der deutsche Pädagoge Friedrich Fröbel erkannte dies schon frühzeitig und sowohl die Eröffnung des ersten Kindergartens 1837 als auch die Entwicklung der „Fröbel gifts“ (Fröbels Spielgaben) können auf ihn zurückgeführt werden. Bei Letzteren handelt es sich um Klötze, Perlen und Platten. Auch Wollknäuel und Lehm zählen dazu. Mit diesen können die Kinder gestalterisch aktiv sein, basteln, spielen und vieles mehr. (Kafai, Peppler, Burke, Moore & Glosson, 2010, S. 214) Die Elemente gestaltete Fröbel so, dass Kinder während dem Spiel alltägliche Muster und Formen aus der Natur lernen können. (Resnick, 2007, S. 3)

„In effect, Fröbel was designing for designers - he designed objects that enabled children in his kindergarten to do their own designing.“ (Resnick, 2007, S. 3)

Heute sind Fröbels Spielgaben² nicht nur aus einfachen Materialien wie im frühen 19. Jahrhundert in Verwendung, sie sind auch Grundlage und Inspiration für zahlreiche Initiativen und Produkte unserer heutigen Zeit. Vor allem im Bereich der Vermittlung von grundlegenden Programmierkenntnissen spielen sie eine ganz wesentliche Rolle. (Resnick, 2007, S. 3) So basieren LEGO® Mindstorms®, LOGO, Scratch und zahlreiche andere Produkte in diesem Metier auf den Ideen und Ausprägungen von Fröbels Spielgaben. (Kafai et al., 2010, S. 214)

2.1.2 Konstruktivismus

Der Konstruktivismus hat seine Wurzeln in der konstruktivistischen Philosophie. Zentrale Personen in diesem Kontext sind Bruner, Piaget, Vygotsky und von Glasersfeld. Die zentrale Aussage des Konstruktivismus ist, dass Wissen nicht vom Lehrenden an den Lernenden direkt übergeben werden kann, sondern es sich beim Lernen um einen aktiven Prozess der Konstruktion handelt, bei dem die Lernenden einen aktiven Part in ihrem eigenen Lernen übernehmen müssen. (Hadjerrouit, 2005, S. 45)

In den letzten Jahren hat sich der Konstruktivismus als die grundlegende Lerntheorie nicht nur für den Informatikunterricht etabliert. (Hadjerrouit, 2005, S. 45), (Ben-Ari, 2001, S. 45) Um den Konstruktivismus in die Praxis umzusetzen, benötigt es ein Bündel an pädagogischen Richtlinien und Strategien, die sich mit Konstruktion, kognitiven Fähigkeiten,

² Fröbels Spielgaben sind heute unter anderem unter: <http://www.friedrich-froebel-online.de/s-p-i-e-l-g-a-b-e-n/>, 23.04.2018, erhältlich.

authentischen Aufgaben, verwandten Fällen, Zusammenarbeit und der Integration von Informationstechnologien beschäftigen. (Hadjerrouit, 2005, S. 45 f.)

Zusammengefasst lässt sich der Konstruktivismus wie folgt darstellen: Wissen wird konstruiert und nicht übermittelt, dabei wird die Wissenskonstruktion in die Interessen und die Umgebung der Lernenden integriert. Die Lernenden spielen in der Wissenskonstruktion einen wichtigen und aktiven Part, der durch die soziale Interaktion, die als essentiell betrachtet wird, mit anderen Lernenden gestärkt wird. (Li, 2001, S. 433)

Die fortschreitende Entwicklung der Informationstechnologie bringt neue Herausforderungen für zukünftige Arbeitskräfte mit sich. Es sind Kreativität, kritisches Denken zum Lösen von Problemen, kombiniert mit den Fähigkeiten zu kommunizieren und zu kollaborieren erforderlich, um neues Wissen und neue Fähigkeiten zu erschließen. Daraus resultieren jedoch auch Anforderungen an die Ausbildung: Lernumgebungen müssen die Gelegenheiten bieten, die sich ständig ändernde Welt zu verstehen und neues Wissen zu kreieren. (Li, 2001, S. 433)

Es zeigt sich, dass die intrinsische Motivation der Lernenden steigen kann, wenn der persönliche Lernstil der Lernenden im Unterricht berücksichtigt und unterstützt wird. (Larkin & Budny, 2005, S. 1) Die große Herausforderung ist jedoch, den persönlichen Lernstil der Lernenden zu erkennen (Ben-Ari, 2001, S. 50), denn Lernstile sind sehr vielfältig und können so individuell sein, wie ein Fingerabdruck – oberflächlich ähnlich, aber im Detail sehr unterschiedlich. (Larkin & Budny, 2005, S. 2) Um einen Grundstock für lebenslanges Lernen zu legen, bedarf es Lernumgebungen, die selbstgesteuertes Lernen ermöglichen. (Lord, Chen, Nottis, Stefanou, Prince & Stolk, 2010, S. 381)

„To be a lifelong learner includes characteristics consistent with those of self-directed learners, such as being curious, motivated, reflective, analytical, persistent, flexible and independent.“ (Lord et al., 2010, S. 381)

2.1.3 Kognitive Entwicklung nach Piaget

Die Beschreibung der kindlichen kognitiven Entwicklung wird untrennbar mit Jean Piaget, einem Schweizer Entwicklungspsychologen, in Zusammenhang gebracht. Seine kognitive Entwicklungstheorie entstammt einer universal-konstruktivistischen Sichtweise. Kinder konstruieren durch ihr Einwirken auf ihre Umwelt, ihr Erkunden und das bewusste Abbilden (mentale Repräsentation) ihr Wissen selbst. Basierend auf dem biologischen Konzept der Adaptation, entwickeln sich die mentalen Strukturen passend zur Umwelt des Individuums. (Berk, 2005, S. 24 f.)

„Der Verstand des Menschen im Säuglingsalter und in der frühen Kindheit unterscheidet sich von dem des Erwachsenen. So war Piaget der Meinung, dass Säuglinge nicht verstehen können, dass ein Gegenstand oder eine Person, die sich außer Sichtweite befinden (das Lieblingsspielzeug oder sogar die Mutter selbst), auch weiterhin existieren.“ (Berk, 2005, S. 25)

Falsche Schlussfolgerungen, begründet in Fehlern im logischen Denken, werden erst nach und nach durch Wissenszuwachs und stetiges Bemühen des Kindes beseitigt. Es entsteht mit der Zeit ein Gleichgewicht zwischen innerer Struktur und den Informationen aus der individuellen Umwelt. (Berk, 2005, S. 25)

Nach Piaget durchläuft das Kind vier Stadien, die durch qualitativ unterschiedliche Arten des Denkens charakterisiert sind. (Berk, 2005, S. 25)

Berk (2005) fasst die vier Stufen der kognitiven Entwicklung wie folgt zusammen:

1. Sensumotorische (oder auch sensomotorische) Stufe, von der Geburt bis zum Alter von ca. 2 Jahren:

„Der Säugling ‚denkt‘, indem er mit den Augen, den Ohren, den Händen und dem Mund aktiv auf seine Umwelt einwirkt. Infolgedessen entdeckt er Möglichkeiten zur

Lösung sensumotorischer Probleme, etwa an einem Faden zu ziehen, um die Spieluhr zum Klingen zu bringen, oder verstecktes Spielzeug zu finden oder auch Objekte in einen Behälter zu bekommen oder sie wieder herauszunehmen.“ (Berk, 2005, S. 26)

2. Präoperationale Stufe, im Alter von ca. 2 bis ca. 7 Jahren:

„Vorschulkinder verwenden Symbole zur Repräsentation ihrer frühen sensumotorischen Entdeckungen. Die Entwicklung des Symbolspiels fällt in diese Phase. Das Denken entbehrt jedoch noch der Logik der beiden späteren Phasen.“ (Berk, 2005, S. 26)

3. Konkret operationale Stufe, im Alter von ca. 7 bis ca. 11 Jahren:

„Das Denken des Kindes wird prälogisch. Dem Schulkind ist es nun möglich zu verstehen, dass eine bestimmte Menge Limonade oder Spielknete, auch nachdem sich ihr Erscheinungsbild verändert hat, dieselbe bleibt. Objekte werden hierarchisch in Gruppen und Untergruppen geordnet. Das Denken reicht aber immer noch nicht an die Intelligenz des Erwachsenen heran, da es noch nicht formal-abstrakt ist.“ (Berk, 2005, S. 26)

4. Formal operationale Stufe, im Alter von ca. 11 Jahren und darüber:

„Die Fähigkeit zur Abstraktion und zu formalem Denken erlaubt es dem Adoleszenten, in Symbolen zu denken, die sich nicht direkt auf Objekte in der realen Welt beziehen, wie etwa in der fortgeschrittenen Mathematik. Auch können alle Lösungsmöglichkeiten eines wissenschaftlichen Problems in Betracht gezogen werden, nicht nur solche, die auf der Hand liegen.“ (Berk, 2005, S. 26)

2.1.4 Konstruktionismus nach Papert

Neben dem Konstruktivismus und den an ihn angelehnten Lehr- und Lernformen spielt heute vor allem auch der Konstruktionismus (constructionism), der auf dem Konstruktivismus basiert, eine bedeutende Rolle in der IKT-Ausbildung. Den Konstruktionismus begründete Seymour Papert, ein Schüler Jean Piagets – „[...] while I was working with Piaget. In 1964, after five years at Piaget’s Center for Genetic Epistemology in Geneva, I came away impressed by his way of looking at children as the active builders of their own intellectual structures.“ (Papert, 1993, S. 19)

Romeike (2011) erklärt den Zusammenhang zwischen Konstruktivismus und Konstruktionismus folgendermaßen: „Papert greift in seiner Lerntheorie des Konstruktionismus vom Konstruktivismus das Prinzip des Lernens als Konstruktionsprozess auf, fügt aber die Idee hinzu, dass Lernen durch das eigenhändige Konstruieren eines (auch gedanklichen) Artefakts unterstützt wird.“ (Romeike, 2011, S. 356) Hierbei ist es wichtig, dass die Lernenden in die Situation eintauchen und sich mit dem Lerngegenstand verbunden fühlen. (Romeike, 2011, S. 357)

Nach Papert (1993) können junge Kinder sehr gut Sprachen lernen, lernen sie doch von Anbeginn an eine Sprache, ihre Muttersprache. Diesen Gedanken aufgreifend verfolgte er in seiner Arbeit zwei grundlegende Ideen. Zum einen sollen Kinder so mit Computern interagieren können, wie sie eine natürliche Sprache am besten lernen – dadurch, dass sie von der Sprache umgeben sind, etwa in dem Land, in dem diese gesprochen wird und nicht durch Sprachunterricht, fernab dieses Landes. Zum anderen kann diese Kommunikation mit dem Computer zu einer Veränderung des Lernens generell führen. Der Computer kann sowohl ein Mathematik sprechendes als auch ein das Alphabet sprechendes Element sein. Wenn Kinder lernen mit dem Computer zu kommunizieren, dann lernen sie Mathematik als eine natürliche Sprache. (Papert, 1993, S. 6) Es wäre, wie wenn man eine Fremdsprache im betreffenden Ausland lernen würde: Mathematik nach Papert (1993) dann im „Mathland“. (Papert, 1993, S. 6)

Papert (1993) ist der Meinung: „If we really look at the child as a builder [...] all builders need materials to build with.“ (Papert, 1993, S. 7) – und hier ist der Unterschied zu Piaget zu sehen. Papert sagt, dass die langsamere Entwicklung eines speziellen Konzepts nicht in der Komplexität oder Formalität dieser Konzepte begründet liegt, sondern vielmehr darin, dass die umgebende Kultur keine Materialien zur Verfügung stellt, die dieses Konzept einfach und konkret erfassbar machen würden. Darüber hinaus kann es nach Papert (1993) auch vorkommen, dass diese Materialien zwar existieren würden, Kinder aber der Zugang verwehrt wird. (Papert, 1993, S. 7 f.) Wobei seiner Einschätzung nach bei vielen Teilgebieten der Mathematik beides eintritt. (Papert, 1993, S. 8) Um für Kinder die Möglichkeit zu schaffen mit Hilfe geeigneter Materialien das Kommunizieren mit Computern zu erlernen („learn to communicate with computers“ (Papert, 1993, S. 8)), hat er mit seiner Forschungsgruppe daran gearbeitet, entsprechende Umgebungen zu gestalten. (Papert, 1993, S.8) Papert (1993) betont die Wichtigkeit der „objects to think with“ aus seiner eigenen Bildungsbiographie als Mathematiker heraus, wo u. a. Zahnräder eine große Rolle für sein mathematisches Verständnis gespielt haben. (Papert, 1993, S.11), (Bers, 2008, S. 4)

Papert ist der Überzeugung, dass der sinnvolle Einsatz von Technologie das Lernen bereichern und unterstützen kann. (Papert, 1970, S. 1) Im Kontext des lebenslangen Lernens betrachtet er es als Pflicht der Lehrenden, den Lernenden das Lernen, Denken und Spielen beizubringen. (Papert, 1970, S. 2)

Die von ihm entwickelte „Turtle“ sieht Papert als Beispiel eines „computational object to think with“. Kommuniziert werden kann mit der Turtle, welche als abstraktes Objekt am Bildschirm oder auch als physisches Objekt in Form der Boden-Turtle vorliegt, mit der Programmiersprache LOGO welche aller Wahrscheinlichkeit nach die erste eigens für Kinder erschaffene Programmierumgebung darstellt. (Papert, 1993, S. 11), (Sheehan & Read, 2009, S. 1647) Die LOGO-Schildkröte ist heute als Roamer bekannt, siehe Kapitel 4.4.2. Erst die zunehmende Verbreitung von Personal Computern führte dazu, dass die LOGO-Schildkröte als Software angeboten wurde. (Blikstein, 2013, S. 174) Eine noch größere Popularität erlangte LOGO Anfang der 1980er Jahre, als Personal Computer, damals der Apple II, Einzug in Schulen hielten. (McNerney, 2004, S. 328)

Die Grundidee Paperts war, dass Kinder abstrakte geometrische Konzepte leichter lernen, wenn sie diese geometrischen Formen physisch erzeugen können. Der körperzentrierte Ansatz – aus Sicht des Bodenroboters werden die notwendigen Schritte beschrieben um die gewünschte geometrische Form zu erhalten – eignet sich sehr gut für Kinder. (McNerney, 2004, S. 327)

„Logo can be used in different ways: as a programming tool for children to learn problem solving in the sense originally intended by Papert, as a tool to create learning environments in different subjects, and as a medium for providing and processing information generally.“ (Brouwer, Muller & Rietdijk, 2007, S. 444)

Zur Verfolgung von Paperts Ansatz eignen sich Roboter sehr gut, denn sie helfen Lernenden abstrakte Gedanken, Ideen und Formen durch Einsatz von realen Objekten zu erforschen. (Beals & Bers, 2006, S. 343 f.), (Gallagher & Baurerle, 2003, S. 1 f.)

Bereits in den 1970er Jahren hat Radia Perlman mit TORTIS ein „tangible programming system“ entwickelt. (Morgado, Cruz & Kahn, 2006, S. 1903), (Gujberova & Kalaš, 2013, S. 109) Basierend auf LOGO erlaubt TORTIS die Steuerung einer Hardware-Turtle via Befehlen auf Plastikkarten. Das System ließ sich schon damals im Kindergarten einsetzen, auch wenn es nach heutigen Erkenntnissen wohl nicht als altersadäquat zu klassifizieren ist. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 132)

2.1.5 ZPD – Zone of proximal development nach Vygotsky

Vygotskys „Zone of proximal development“ ist definiert als die Entfernung zwischen dem tatsächlichen Entwicklungsniveau, welches durch die unabhängige Problemlösungskompetenz bestimmt ist, und dem potenziellen Entwicklungsniveau, welches sich aus der

Problemlösungs-Kompetenz unter Anleitung eines Erwachsenen oder eines fähigeren Peers ableiten lässt. (Beals & Bers, 2006, S. 345), (Silva, Barros, Azevedo, Silva & Gonçalves, 2008, S. 132) Vereinfacht ausgedrückt kann die „Zone of proximal development“ auch als die Distanz zwischen dem Handeln einer Person ohne Hilfe und dem Handeln einer Person mit Hilfe angesehen werden. (Verenikina, 2010, S. 18) Die helfende Person agiert hierbei unterstützend und als „Wissensbank“, die Strategien, Prozeduren, Aussagen und Interpretationen zur Verfügung stellt. (Meadows, 1998, S. 7)

Beals & Bers (2006) untersuchten den Einfluss der „Zone of proximal development“ auf das Lernen von Kindern und verglichen zwei Gruppen: Zum einen eine Gruppe Kinder, die mit LEGO® Mindstorms® arbeiteten und zum anderen eine Gruppe Eltern mit Kindern, die ebenfalls mit LEGO® Mindstorms® arbeiteten. Die Gruppe Eltern-Kinder schnitt bei den Tests und Evaluierungen der fertigen Roboter besser ab, als die Gruppe der Kinder. (Beals & Bers, 2006, S. 358) Es zeigte sich, dass die Konstrukte der Eltern-Kinder-Gruppe deutlich komplexer und komplizierter waren, als die der Gruppe Kinder. Interessant zu beobachten war jedoch, dass die Kinder in der Gruppe Kinder ihr Wissen stärker erweiterten und aufbauten, als die Kinder in der Gruppe Eltern-Kinder, wo die Eltern ebenso einen Wissenszuwachs verzeichneten, aber die Kinder selbst einen geringeren. (Beals & Bers, 2006, S. 359) Die Kinder aus der Gruppe Kinder waren eher in der Lage Aufgabenstellungen selbsttätig zu lösen, als die Kinder in der Gruppe Eltern-Kinder. Während den Vorbereitungen hatten jedoch beide Gruppen die gleichen Chancen sich Hilfe von den Lehrenden zu holen. (Beals & Bers, 2006, S. 360) Kinder aus der Gruppe Kinder konnten einfache Aufgaben selbst lösen, die Kinder aus der Gruppe Eltern-Kinder konnten zwar über die Programmierung, die Musik oder Konzepte plaudern, waren aber nicht in der Lage, die einfachen Aufgaben selbst zu lösen. (Beals & Bers, 2006, S. 361) Beals & Bers (2006) schlussfolgern, dass dosierte Hilfestellungen helfen, die eigenen Fähigkeiten weiter auszubauen, wenn aber Hilfestellungen zu kompliziert und zu weit entfernt sind von dem eigenen Wissen, dann lernt man tatsächlich weniger. (Beals & Bers, 2006, S. 361) Nebenbei bemerkten sie auch, dass in der Gruppe Eltern-Kinder viele Eltern versuchten, sich selbst weiter zu entwickeln und noch kompliziertere Aufgaben zu lösen um selbst ihren eigenen Lernfortschritt zu steigern. Dabei wurden die Kinder überfordert, so dass sie sich zunehmend aus dem Lernen zurückzogen. Es gilt in der Praxis richtig dosierte Hilfestellungen zu geben um den Lernfortschritt zu fördern. (Beals & Bers, 2006, S. 362 f.)

Es ist für Lehrende eine große Herausforderung, Vygotskys Theorie in die Praxis umzusetzen – Gruppenarbeiten zu fördern, die von Individuen unterschiedlicher Herkunft (Kultur, soziales Umfeld, Bildung) bestritten werden und zum gegenseitigen miteinander Lernen motivieren sollen. (Stringer, 1998, S. 19)

2.2 Frühe Technikbildung im Kontext des lebenslangen Lernens, mit Schwerpunkt Österreich

„Kinder sind neugierige und forschende Persönlichkeiten. Ihr hohes Lernpotenzial stellt die Basis für ihre gesamte Bildungsbiografie dar. Die Wertschätzung und Förderung früher Bildungsprozesse entsprechen einem pädagogischen Anspruch und einem gesellschaftspolitischen Auftrag.“ (Hartmann, Bäck, Hajszan, Hartel, Kneidinger, Stoll, Bair, Baumann, Grötschnig, Hutz, Minich, Salzmann, Steger, Strasser & Thoma, 2009, S. 1)

Mit der Entwicklung des „Bundesländerübergreifenden BildungsRahmenPlans für elementare Bildungseinrichtungen in Österreich“ haben sich die einzelnen Landesregierungen der neun Bundesländer in Österreich zum Bildungsauftrag der elementaren Bildungseinrichtungen bekannt. Das Kind wird hierbei als kompetentes Individuum betrachtet, das seine Entwicklung aktiv mitgestaltet. (Hartmann et al., 2009, S. 1)

Es ist dem Kind ein intrinsisches Bedürfnis sich mit seiner Umwelt auseinanderzusetzen, zu versuchen sie zu begreifen und dies als Chance für die Entwicklung von Kompetenzen zu nutzen. Neben fachlichen Kompetenzen prägen sich auch Selbstkompetenz,

Sozialkompetenz und lernmethodische Kompetenz aus. (Hartmann et al., 2009, S. 6) Die Entwicklung letzterer ist Basis für alle Lernprozesse im weiteren Leben des Menschen. (Hartmann et al., 2009, S. 7), (McDermott, Leigh & Perry, 2002)

Es gilt in Folge den Begriff „Kompetenz“ zu definieren. Nach dem BildungsRahmenPlan 2009 für elementare Bildungseinrichtungen in Österreich, ist unter Kompetenz „ein Netzwerk von Kenntnissen, Fähigkeiten und Fertigkeiten, Strategien und Routinen“ zu verstehen, welches von Menschen benötigt wird, um neben ihrer Lernmotivation auch in weiteren „unterschiedlichen Situationen handlungsfähig zu sein“. (Hartmann et al., 2009, S. 6)

Jung (2010) definiert Kompetenz als „Befähigung zur Bewältigung komplexer Situationen“. Eine Kompetenz ist eine „menschliche Eigenschaft, in Abhängigkeit von den individuellen Lebensbedingungen kognitive, soziale und verhältnismäßige Fähigkeiten so zu organisieren und einzusetzen, dass Ziele, Interessen und Wünsche zu verwirklichen sind“. (Jung, 2010, S. 26)

In diesem Kontext kommt dem Kindergarten eine ganz besondere Bedeutung zu. Er ist ein „freier“ Raum, in dem Bildung ohne Notengebung stattfindet und jedes Kind Wissen selbst erlangen kann. (Dräger, 2008, S. 148 f.), (Schachtner, Roth & Frankl, 2005, S. 51), (Janus & Brinkman, 2010, S. 25) Um die Nachwuchsförderung von technischen und naturwissenschaftlichen Arbeitskräften voranzutreiben empfehlen Prenzel, Reiss & Hasselhorn (2009) eine Schwerpunktsetzung „im Aufbau von Medienwissen und computerbezogenen Kompetenzen“. (Prenzel, Reiss & Hasselhorn, 2009, S. 21)

Die Hauptaufgabe der Schule und damit vor allem auch der elementaren Bildungseinrichtungen liegt nach Nievergelt (1999) darin, das „gedankliche Rüstzeug zu vermitteln, das uns lebenslang begleiten kann“. (Nievergelt, 1999, S. 374) Er vertritt die These, dass „Programmieren zum allgemeinen Gedankengut einer modernen Industrie- und Dienstleistungsgesellschaft gehört“. (Nievergelt, 1999, S. 374)

Informations- und Kommunikationstechnologien (IKT) sind aus dem Alltag nicht mehr wegzudenken und eine elementare zeitgemäße Bildung umfasst daher auch die Förderung kindlicher Medienkompetenzen. (Hartmann et al., 2009, S. 15) „Dies befähigt Kinder, unterschiedliche Medien zunehmend selbstgesteuert und kritisch zu nutzen.“ (Hartmann et al., 2009, S. 15) Die wichtigsten Handlungskompetenzen für lebenslanges Lernen umfassen „grundlegende naturwissenschaftlich-technische sowie mathematische Kompetenzen“. (Hartmann et al., 2009, S. 20) Die Kompetenzen werden im Verlauf der Schulzeit aufgebaut, wobei den sehr frühen Schuljahren eine besondere Bedeutung zukommt. Zu berücksichtigen gilt es insbesondere, dass „Defizite in der Lerngeschichte selbst durch eine hohe Intelligenz nicht hinreichend kompensiert werden“ können.³ (Prenzel, Reiss & Hasselhorn, 2009, S. 17)

Weiter begann in den letzten Jahren ein Wandel in der Beurteilung der Leistungen von Lernenden. Standen früher noch die klassischen Testaspekte im Vordergrund, rücken nun zunehmend Bewertungsaspekte in Bezug auf Kompetenzen in den Fokus. Während die Test-Kultur noch eher einem behavioristischen Ansatz zuzuordnen ist (es zählt vor allem das Endergebnis, der Prozess dorthin ist aus Beurteilungssicht irrelevant), ist in der Bewertungskultur ein Bezug zu kognitiven Ansätzen zu finden (der Prozess zur Ergebnisfindung ist hierbei durchaus von Interesse). (Baartman, Bastiaens, Kirschner & van der Vleuten, 2007, S. 116) Das führt zu einer Veränderung des Bildungssystems, von einer humanistischen Bildung – wo der Fokus auf dem Individuum der Lernenden liegt – hin zu einer kompetenzorientierten. (Neumann, Fischer & Kauertz, 2010, S. 545 f.)

Parallel dazu ging die Informatik im Laufe der letzten Jahre von einer Ingenieurwissenschaft bzw. Formalwissenschaft zu einer Basiswissenschaft der Gesellschaft über, die sich durch

³ Vorschulische Mengen-Zahl-Kompetenzen sind ein maßgebliches Vorhersagekriterium über die mathematische Leistung in der Volksschulzeit. „Insbesondere haben Kinder, die bereits vor Schuleintritt mit mathematikbezogenen Aufgaben erfolgreich umgehen können, einen über Jahre relativ stabilen Vorsprung vor ihren Altersgenossen.“ (Prenzel, Reiss & Hasselhorn, 2009, S. 17)

ihre zahlreichen Methoden und Paradigmen der Modellierung, Abstraktion und Konstruktion sowie ihren Einfluss auf andere Wissenschaftsdomänen so wertvoll für die Allgemeinbildung macht und in weiterer Folge zur Beherrschung von Komplexität beiträgt. (Schubert & Schwill, 2011, S. 117 f.) Es erscheint daher wichtig, die Informatik im Kontext der Kompetenzentwicklung zu betrachten, wie das Mulley & Zuliani (2013) z.B. mit digi.komp4⁴, einem Kompetenzmodell für digitale Kompetenzen über die Kinder nach Besuch der Volksschule verfügen sollten, gemacht haben. (Mulley & Zuliani, 2013, S. 3) Bis zu einem Alter von 12 Jahren lernen die meisten Kinder überdurchschnittlich schnell und ohne allzu großen Aufwand. Danach lässt die Lernleistung nach, vor allem, wenn man nicht mehr regelmäßig lernt und so sein Gedächtnis auf Neues trainiert. (Janacsek, Fiser & Nemeth, 2012, S. 501)

„Gesellschaftliche, ökonomische und technologische Veränderungen erfordern die Entwicklung von Lernkompetenzen als Voraussetzung für ein lebenslanges selbstorganisiertes Lernen.“ (Hoidn, 2011, S. 1) Hoidn (2011) zeigt auf, dass Wissen als „individuell, sozial, materiell und kulturell“ verteilt zu betrachten ist. Daher ist für die Wissenskonstruktion und die Fähigkeit lernen zu können, auf motivational-emotionale Prozesse, sowie auch auf kulturelle, soziale und mediale Interaktionsprozesse Rücksicht zu nehmen. (Hoidn, 2011, S. 8) Selbstorganisiertes Lernen ist ein aktives, sozial orientiertes reflexives Problemlösen in der Gruppe, die sich den Lernprozess selbst plant, gestaltet, durchführt und bewertet. (Hoidn, 2011, S. 14)

Eine große Herausforderung stellt die Informatik dar, wo es gilt ein gemeinsames Verständnis zu schaffen, welche Konzepte der Informatik unbedingt in den Schulunterricht eingebracht werden müssen. (Dagiene & Futschek, 2010, S. 8) Der Vergleich der Informatik-Standards und der Schul-Curricula zeigt nur geringe Übereinstimmungen auf, sogar die informatischen Schlüsselkompetenzen Algorithmen und Programmierung sind in den Standards und Schul-Curricula oftmals nicht enthalten. (Dagiene & Futschek, 2010, S. 14)

Hartmann, Bäck, Hajszan, Hartel, Kneidinger, Marek, Pfohl-Chalaupek & Stoll (2010) verfeinern in ihren „vertiefenden Ausführungen zum bundesländerübergreifenden BildungsRahmenPlan“ (in Österreich), welche Fähigkeiten, Fertigkeiten und Kompetenzen Kinder im Bereich der IKT durch die elementare Bildung mitbekommen sollen. Dazu zählen u.a. „eigene Anliegen und Interessen mit Symbolen, Schrift sowie Informations- und Kommunikationstechnologien ausdrücken [...] unterschiedliche Medien selbständig nutzen und als Möglichkeit zur Informationsbeschaffung einsetzen [...] Den Unterschied zwischen Realität und Fiktion erkennen und adäquat darauf reagieren.“ (Hartmann et al., 2010, S. 29)

Darüber hinaus postulieren sie: „Das wichtigste Anliegen einer zeitgemäßen Medienbildung in elementaren Bildungseinrichtungen ist es, für alle Kinder Grundlagen einer gleichberechtigten Teilhabe an der Wissens- und Informationsgesellschaft zu schaffen.“ (Hartmann et al., 2010, S. 29)

Doch gerade in diesem Bereich trifft man auf ein Problem, das nach Prenzel, Reiss & Hasselhorn (2009) zumindest in Deutschland nicht trivial gelöst werden dürfte. „Besondere Anstrengungen sind freilich auch in der Ausbildung des pädagogischen Personals für den vorschulischen Bereich nötig, [...] [da, Anm. d. Verf.] der Schulabschluss, der für eine Ausbildung zur Erzieherin oder zum Erzieher erforderlich ist, nur eine marginale Vertrautheit mit mathematischen, naturwissenschaftlichen und technischen Themen gewährleistet.“ (Prenzel, Reiss & Hasselhorn, 2009, S. 44) Ähnliches ist in Österreich zu beobachten, wie z.B. Mittermeier, Bischof & Hodnigg (2010, S. 143) berichten. Aber auch z.B. in Hongkong haben Kindergärten mit ähnlichen Herausforderungen zu kämpfen. (Han, 2003, S. 49)

⁴<https://www.edugroup.at/praxis/portale/digitale-kompetenzen/digikomp4-volksschule/kompetenzmodell.html>, 29.12.2018

Ebenso arbeitet Dräger (2008) heraus, dass die Kindergartenarbeit, auch aufgrund des Interesses des pädagogischen Personals, eher im kreativen und künstlerischen als im mathematisch-naturwissenschaftlichen Bereich angesiedelt ist. Die Förderung des Interesses im mathematisch-naturwissenschaftlichen Bereich wird in der Ausbildung des pädagogischen Personals oftmals unterlassen. (Dräger, 2008, S. 148) Zahlreiche Befunde in der Literatur zeigen auf, dass dies ein weit verbreitetes Phänomen ist. (Turja, 2009) (Baldu, 2002), (Fagan, 2011), (Rosen, 2006), (Ntuli & Kyei-Blankson, 2011)

Weiter belegen Studien, dass es nicht ausreicht, Kindern im Kindergartenalter die Möglichkeit zu bieten Computer zu nutzen, sondern sich Kindergartenpädagoginnen und -pädagogen in einem Lehr-Lern-Setting mit Computerunterstützung aktiv mit den Kindern beschäftigen sollten. Denn nur so können Kinder von den Vorteilen eines Computereinsatzes, z.B. Steigerung der kognitiven Leistung, profitieren. (Nir-Gal & Klein, 2004), (Berndt, 2009) Ein Einsatz von IKT in der Lehre darf nicht als „Betreuungs-Ersatz“ zur stillen Beschäftigung der Kinder betrachtet werden. (Rivera, Galarza, Entz & Tharp, 2002, S. 202) (Herper & Hinz, 2010, S. 354) In Lehr-Lern-Szenarien mit IKT sollte viel Wert darauf gelegt werden, dass die Lernenden nicht nur die Ergebnisse auf ihre Richtigkeit überprüfen, sondern auch die Prozesse kennen und verfolgen, die die Ergebnisse erzeugten. (Berndt, 2009, S. 241) (Han, 2003, S. 51)

Dies erfordert jedoch vielerorts erst eine adäquate technische Ausstattung der Unterrichtsräumlichkeiten (Martin, Seevers & Crawford, 2002) und die Schaffung von Organisationsstrukturen und Verfügbarkeiten von administrativem Personal. Beides wird zu einem kollaborativen Einsatz von IKT in einer Schule / einem Kindergarten benötigt. (Leclerc, 2006, S. 4 ff.) Bildungseinrichtungen sind zum Teil mit komplexen und kostenintensiven Anforderungen konfrontiert, um eine zeitgemäße IKT-Ausbildung und -Unterstützung im Unterricht zu gewährleisten. (Herper & Hinz, 2010, S. 353 f.)

2.3 IKT-Aus- und Weiterbildung von Kindergartenpädagoginnen und -pädagogen

Bereits im Jahr 2000 haben Chambers & Tromp (2000) aufgezeigt, dass IKT-Kompetenzen für Lehrkräfte nicht nur sinnvoll für deren eigene Arbeit sind, sondern auch in der Forschung, Weiterbildung, Administration und vor allem auch im Unterricht als Hilfsmittel eine große Rolle spielen. (Chambers & Tromp, 2000, S. 1238), (Gibson, 2010)

Mit Initiativen, Best-Practice-Studien und exemplarischen Ausbildungsszenarien wird die Integration von IKT in die Aktivitäten von Kindergärten bereits in der Ausbildung der Pädagoginnen und Pädagogen gefördert. (Nir-Gal & Klein, 2004), (Han, 2003), (Alimisis, 2008), (Ma, Prejean & Williams, 2011), (Dack, Ford & Prejean, 2006), (Bell & Crawford, 2001) Dabei werden die Forderungen der Pädagoginnen und Pädagogen nach mehr Vorbereitungszeit und technischer Unterstützung bei selbiger lauter. (Ntuli & Kyei-Blankson, 2011) Wünschenswert wären weiter IKT-Coaches, die Pädagoginnen und Pädagogen zu einem niederschweligen Einstieg beim Einsatz von IKT im Unterricht verhelfen. (Sugar, 2005), (Brown & Warschauer, 2006)

„The challenge to education is to transform information technologies into learning technologies.“ (Brouwer, Muller & Rietdijk, 2007, S. 442)

Die Möglichkeiten für einen IKT-Einsatz in Lehr-Lern-Szenarien sind breitgefächert. Als Lernmaterialien können Standardsoftware (siehe u.a. Berndt (2009)), Computerspiele (siehe u.a. Verenikina, Herrington, Peterson & Mantei (2010)), Roboter (siehe u.a. Okita, Schwartz, Shibata & Tokuda (2005)), virtuelle Welten (siehe u.a. Bers, Beals, Chau, Satoh & Khan (2010)), das Internet (siehe u.a. Schachtner, Roth & Frankl (2005)) bis hin zu sog. smarten Kindergärten (SmartKG) (Chen, Muntz, Yuen, Locher, Park & Srivastava, 2002) sowie viele weitere Hard- und Softwareprodukte zum Einsatz kommen. Dabei ist vor allem auf die Alltagstauglichkeit (Installation, Einsatz, Wartung) und die Bedienbarkeit der Produkte, im

Speziellen der Software, Wert zu legen. (Nikiforidou & Pange, 2010), (Rosen & Jaruszewicz, 2007)

Fagan (2011) hält fest, dass es nicht von Nöten ist, dass Kindergartenpädagoginnen und -pädagogen ausgewiesene IKT-Expertinnen und -Experten werden, jedoch sollten sie entsprechendes Interesse und die Bereitschaft zur Weiterbildung in diesem Bereich mitbringen, um eine adäquate Pädagogik im 21. Jahrhunderts anbieten zu können. (Fagan, 2011, S. 2959), (Alexander, Digiovanni & Rosen, 2011), (DeClue, 2008) Ein Grundverständnis ist aber unumgänglich um den Kindern, Schülerinnen und Schülern mögliche Gefahren im Umgang mit IT-Systemen aufzeigen und im Unterricht präventiv agieren zu können. Eine frühzeitige Einbindung von Lehr-Lern-Inhalten zum Thema Informationssicherheit und der richtige Umgang mit IT (vor allem in Hinblick auf Datenschutz, Persönlichkeitsrechte, etc.) werden hierbei von Nöten. (Alexander & Rackley, 2005) Die idealtypische Verknüpfung von Theorie und Praxis erscheint unumgänglich um mit der Weiterentwicklung der IKT Schritt halten zu können. (Baumert & Kunter, 2006, S. 506)

2.3.1 Integration von IKT in Lehr-Lern-Szenarien

Kalaš (2010) untersucht, welche Aspekte zur Integration von IKT in den elementaren Bildungsbereich eine Rolle spielen. Er behandelt hierbei die Themen wie der Integrationsprozess von IKT initiiert werden kann, welche Werkzeuge der IKT und wie diese Werkzeuge eingesetzt werden können, sowie welchen Herausforderungen dabei die Lehrenden und die Eltern gegenüberstehen. (Kalaš, 2010, S. 51 ff.)

Die meisten Lehr-Lern-Konzepte, die sich mit einem IKT-Einsatz im elementaren Bildungsbereich beschäftigen, folgen dem konstruktivistischen Lernparadigma. (Alexander, Digiovanni & Rosen, 2011, S. 4476) Als geeignete Methode beim Einsatz von IKT in der Lehre wird Inquiry-Based-Learning empfohlen. Von den Lernenden sollen durch die Beantwortung von Fragestellungen und durch das Lösen von Aufgaben der Lerngegenstand erkundet sowie die Problemlösungskompetenzen und individuelle soziale Kompetenzen erworben und vertieft werden. (Wang, Kinzie, McGuire & Pan, 2008)

Beim Problem-Based-Learning steht vor allem das Problem an sich im Fokus des Interesses, da es dieses zu lösen gilt. Im Gegensatz dazu zeigt das Project-Based-Learning eine stärkere Fokussierung auf das Endergebnis, als auf den Problemlösungsprozess. (Ford, Dack & Prejean, 2006, S. 1) Zwei der größten Vorteile des Project-Based-Learning sieht Eguchi (2007) darin, dass Lernenden-zentriert gelehrt wird und dass sich die Lernenden eigene Ziele setzen können. (Eguchi, 2007, S. 2547 f.)

Ein wesentlicher Unterschied zwischen Inquiry-Based-Learning und Problem-Based-Learning ist, dass beim Problem-Based-Learning die Frage- und Problemstellungen gewöhnlich von den Lehrerinnen und Lehrern gestellt werden, wohingegen beim Inquiry-Based-Learning auch die Lernenden Fragen und Probleme aufwerfen können, die kollaborativ im Team bearbeitet werden. (Wang et al., 2008, S. 2) Zu beachten ist jedoch, dass die zum Einsatz kommenden „Probleme“, die gelöst werden sollen, einen Bezug zur Lebenswelt der Lernenden aufweisen müssen, denn nur so kann nachhaltig die Motivation und der Lernerfolg hochgehalten werden. (Romeike, 2011, S. 366 f.)

Eine weitere Methode für IKT-basierte Lehr-Lern-Szenarien stellt Discovery Learning dar. Hierbei lernen Menschen durch Interaktionen mit der Umwelt sowie durch Manipulation von Objekten. Weiters stellen die Lernenden Fragen, zeigen Kontroversen auf und führen Experimente durch. Diese Art des Lernens eignet sich sehr gut für technologiebasierte Inhalte wie Robotik. (Sullivan & Moriarty, 2009, S. 112) Mit geeigneter Fragestellung der Lehrenden sollen die Lernenden durch den entdeckenden Lernprozess geführt und begleitet werden. (Sullivan & Moriarty, 2009, S. 113)

Viele Lehrende sehen in derartigen Lehr-Lern-Szenarien den Nachteil, dass die oben genannten Lehr-Lern-Methoden deutlich mehr Unterrichtszeit benötigen als herkömmliche Lehr-Lern-Methoden. (Sullivan & Moriarty, 2009, S. 114) Sullivan & Moriarty (2009) zeigen

jedoch auf, dass sich eine persönliche Auseinandersetzung mit neuen Lehr-Lern-Methoden und Ausprobieren dieser positiv auf die Unterrichtsgestaltung auswirkt. Lehrende erkennen durch das Ausprobieren eher die Vorteile und sind bereit, trotz einem zeitlichen Mehraufwand, diese Methoden im Unterricht einzusetzen. (Sullivan & Moriarty, 2009, S. 124 ff.) Baumert & Kunter (2006) sehen die Herstellung einer konstruktiv-unterstützenden Lernumgebung sowohl als eine Basisdimension der Unterrichtsqualität, unabhängig von einem IKT-Einsatz im Unterricht, als auch als eine der zentralen Herausforderungen an Pädagoginnen und Pädagogen sowie Lehrkräfte in der heutigen Zeit an. (Baumert & Kunter, 2006, S. 488)

2.3.2 Herausforderungen

Es gilt, neben der eigenen Fachkompetenz, zusätzliche Kompetenzen im IKT-Bereich aufzubauen, sowie Methodenkompetenzen auf Basis des Konstruktivismus bzw. seiner daraus abzuleitenden Lehr-Lern-Formen auszubilden. (Gibson, 2010) Vor allem aber verlangt die Weiterbildung von Lehrkräften der Informatik erhöhte Aufmerksamkeit – der rasante Wechsel / die rasche Weiterentwicklung von Technologien erfordert eine ständige Auseinandersetzung mit den aktuellen Entwicklungen sowie eine fortlaufende Gestaltung von adäquaten Lehr- und Lernmaterialien. (Humbert & Pasternak, 2010) Informatikerinnen und Informatikern soll im Rahmen ihrer Ausbildung ermöglicht werden, sich mit der Thematik Lehre und Ausbildung zu beschäftigen, denn nur so können auch zeitgemäße computergestützte Angebote entwickelt und zur Verfügung gestellt werden. (Papastergiou, 2006, S. 6 f.)

Nach Lara & Whittier (2004) steht die ältere Generation an Lehrkräften der Herausforderung des Wechsels des Unterrichtsstils hin zu einem selbstbestimmten, konstruktivistischen Lernparadigma skeptisch, teilweise sogar ängstlich gegenüber – die Furcht die Kontrolle über die Gruppe zu verlieren spielt hier eine Rolle. (Lara & Whittier, 2004, S. 3), (Markovac & Rogulja, 2009, S. 6) Es bleibt zu hoffen, dass entsprechend ausgebildete junge Lehrkräfte ihr Wissen einbringen dürfen und ältere daran partizipieren können.

Im Gegensatz zu Chambers & Tromp (2000), Gibson (2010) und Nir-Gal & Klein (2004), die IKT und Informatik als Werkzeug zur Bereicherung des Unterrichts ansehen, fordert u.a. Hadjerrouit (2009), dass vor allem der Informatikunterricht selbst überarbeitet und neugestaltet werden muss. IKT darf nicht länger nur als Werkzeug für andere Disziplinen betrachtet werden, sondern im Unterricht müssen klar die konzeptuellen Elemente und die Grundlagen der Informatik vermittelt werden. (Hadjerrouit, 2009, S. 227 f.) Hadjerrouit (2009, S. 227 f.) sieht das Problem und die Herausforderung darin, dass Lehrende der Informatik im sekundären Bildungssektor oftmals über unzureichende Kompetenzen in diesem Fachgebiet verfügen und der Unterricht in seiner organisatorischen Einbettung in die Bildungslandschaft zumeist mangelnde Berücksichtigung findet. Um hiervon abzukommen hebt Micheuz (2009) hervor, dass gerade der Informatikunterricht mit sehr gutem Beispiel bei der Integration von Technologien und neuen Lehr-Lern-Methoden vorangehen soll. (Micheuz, 2009, S. 3 & S. 11)

Eine der großen Herausforderungen – gerade im Bereich der Vermittlung von Informatik – stellt die Heterogenität des Vorwissens der Lernenden dar. (Knobelsdorf, 2009, S. 222 f.) Selbstbestimmtes Lernen, angeleitet von einem Coach oder einer oder eines moderierenden Lehrenden, soll es den Lernenden ermöglichen, an individuelles Vorwissen anzuknüpfen und sich individuell weiter zu entwickeln. (Knobelsdorf, 2009, S. 226 ff.)

Der Bereich der Software Entwicklung ist für Romeike (2011) der kreativste Bereich der Informatik (Romeike, 2011, S. 358) und gehört daher unbedingt in die Lehr-Lern-Szenarien mit einbezogen. Schubert & Schwill (2011, S. 31 f.) sprechen vor allem von Problem-, Handlungs- und Anwendungsorientierung, die im Rahmen der Auseinandersetzung mit Informatik trainiert und ausgeprägt werden sollen. (Siehe auch (Knobelsdorf, 2009, S. 226), (Habermann, Muller & Averbuch, 2008, S. 190) oder (Spannagel & Zendler, 2008, S. 28))

Somit wird hier klar deutlich, dass Informatik-Lehr-Lern-Szenarien weit mehr sein sollten, als eine reine „Tool-Schulung“. (Nievergelt, 1999, S. 374)

Diese Implikationen gelten auch für die frühe Technikbildung in Informatik. Kinder verfügen bereits im Kindergarten über heterogene Vorkenntnisse, wie die Autorin der gegenständlichen Arbeit in ihrer Vorstudie (siehe Kapitel 1.4) feststellte, wohingegen Kindergartenpädagoginnen und -pädagogen teilweise keine oder zu wenige IKT-Kenntnisse innehaben, um diese Kindern vermitteln zu können. Unbeachtet dessen, dass die meisten Kindergärten über keine adäquate technische Ausstattung für eine Integration von IKT in den Kindergarten-Alltag mit Kindern verfügen.

Die Herausforderungen fasst Kalaš (2013) wie folgt zusammen:

- Mängel in der Ausbildung der Lehrenden,
- Mangelndes Vertrauen in die pädagogischen Potentiale von IKT und in die eigenen IKT-Fähigkeiten der Lehrenden,
- Mangelnde Unterstützung für die Lehrenden bei technischen Problemen und pädagogischen Fragestellungen zur Integration von IKT in den Lehr-Lern-Alltag,
- Fehlende Lehr-Lern-Materialien für den Lehr-Lern-Alltag mit IKT-Unterstützung,
- Fehlender fortwährender Einsatz von IKT in den Lehr-Lern-Alltag.

(Kalaš, 2013, S. 223)

2.4 Lernen & Spiel – Lernen durch Spiel

In der Literatur wird dem Spiel eine sehr große Bedeutung für das Lernen (nicht nur des Kindes) und die Entwicklung von Kompetenzen zugeschrieben. Spielfreude, unermüdliche Neugierde oder auch die intrinsische Motivation alles begreifen und kennenlernen zu wollen wird bei Hartmann et al. (2009) als „Spielfähigkeit“ bezeichnet. Unterstrichen werden kann diese Aussage mit Hilfe von Rapeepisarn, Wong, Fung & Depickere (2006), die konstatieren, dass Spiel ein Hilfsmittel sein kann, eine rationalere und metaphorischere Art zu Denken auszubilden. (Rapeepisarn et al., 2006, S. 28) Hartmann et al. (2009) bekräftigen diese Aussagen und präzisieren sie sogar noch:

„Die freie Wahl von Spielmaterialien, Spielpartnerinnen und -partnern sowie Spielaktivitäten fördert die Selbstbestimmung der Kinder und bewirkt hohe intrinsische Motivation für ihre Lernprozesse.“ (Hartmann et al., 2009, S. 8)

Nach Rapeepisarn et al. (2006) erscheint das Spiel als ein natürliches und universelles Hilfsmittel zum Lernen, welches gleichwohl für Kinder wie auch Erwachsene geeignet ist. Durch Spielen können Menschen Fähigkeiten und Fertigkeiten erlangen, ohne dies bewusst zu tun und dabei den natürlichsten Weg (des Lernens) bestreiten. (Rapeepisarn et al., 2006, S. 29) Diesem Gedankengang folgend, gilt es in weiterer Folge die beiden Begriffe „learn through play“ und „edutainment“ zu differenzieren. Während unter „learn through play“ alle (Lern-)Aktivitäten subsummiert werden, die spielerisch erfolgen, liegt bei Angeboten des „edutainments“ der Schwerpunkt auf Aktivitäten, bei denen Computer, im Speziellen Computerspiele, zum Einsatz kommen. (Rapeepisarn et al., 2006, S. 28) Dem „edutainment“ liegt ein eher passives Lernen zugrunde, es geht hierbei um Aktivitäten, die dem Lernenden von anderen bereitgestellt werden. Dem gegenüber steht das „learn through play“, Lernen durch Spiel oder spielerisches Lernen, welches Aktivitäten umfasst, die Lernende intrinsisch motiviert ausführen. (Rapeepisarn et al., 2006, S. 31)

Nach Hartmann et al. (2009) steigert die Spielfreude die Lernmotivation, so dass das Lernen durch Spiel als die wichtigste und primär empfohlene Lernform gilt. Dieses Bekenntnis des österreichischen BildungsRahmenPlans (2009) „stellt einen wichtigen Baustein zum erfolgreichen Übergang zwischen den Bildungsinstitutionen dar.“ (Hartmann et al., 2009, S. 1)

Wie bei allen anderen Lernformen entscheiden auch beim „Lernen durch Spiel“ die geeigneten Lernmaterialien über den Erfolg der Lernenden. Im Kontext des „Lernens durch Spiel“ spielen „Hands-on“-Materialien – also Materialien mit der Möglichkeit zur direkten Manipulation der Objekte und zum Angreifen der physischen Repräsentationen (Eguchi & Uribe, 2009, S. 2129) – eine große Rolle, wobei nach Bærendsen, Jessen & Nielsen (2009) durch den Einsatz von Technologien aus Objekten „intelligente Lernmaterialien“ geschaffen werden können. „Intelligente Lernmaterialien“ ermöglichen durch den Einsatz von fortgeschrittenen digitalen Technologien und Robotik, im Vergleich zu herkömmlichen Materialien, physisch fassbare („Hands-on“)-Produkte mit Eigenschaften zu erschaffen, die durch induktives Denken unter Anwendung von Beobachten, „Trial and Error“ und Testen von Hypothesen erkundet und erlernt werden können. Dies stellt den Autorinnen und Autoren zufolge einen sehr natürlichen Weg des Lernens dar, welcher zur konstruktivistischen Lerntheorie passt. Das bedeutet, dass schwer erklärbare und schwer zu verstehende abstrakte Konzepte, Modelle und Theorien für die Lernenden greifbar und konkret werden. (Bærendsen, Jessen & Nielsen, 2009, S. 400) Nach Wyeth (2006) sollten Spielzeuge flexibel, veränderbar und tragbar sein, um aus ihnen immer wieder neue Gegenstände zum Spielen zu formen und so auch die Weiterentwicklung zu fördern. Ebenso sollten diese Spielzeuge soziale und emotionale Verbindungen ermöglichen und so in den Alltag und die Umwelt eingebettet werden können. (Wyeth, 2006, S. 1228)

Was wir lernen und wie wir lernen hängt von unserem Vorwissen und unseren Vorerfahrungen ab. Schülerinnen und Schüler sollen in der Schule die Möglichkeit bekommen, Probleme auf verschiedene Arten, mit ihrem individuellen Vorwissen und ihren individuellen Fertigkeiten, zu lösen, um bei diesem Prozess des Problemlösens neue Erfahrungen und neues Wissen zu gewinnen. Hierbei hat der Einsatz der großen Menge an Wissen und Erfahrungen, die die Kinder in informellen Lernsituationen außerhalb der Schule gewonnen haben, sehr großen Wert. (Bærendsen, Jessen & Nielsen, 2009, S. 408)

Turja (2009b) setzt die zentralen Aktivitäten, die zur Technikbildung in der frühen Kindheit beitragen, wie folgt in Beziehung:

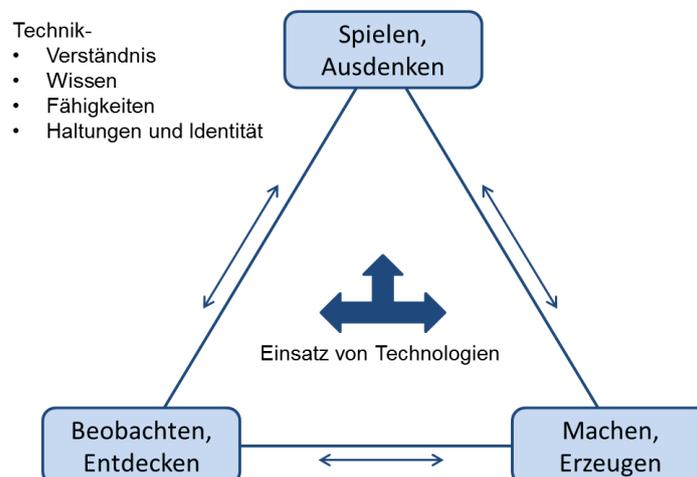


Abbildung 2: Die Hauptaktivitäten in der Technikbildung, eigene Darstellung nach (Turja, 2009b, S. 7)

Beals & Bers (2006) beobachteten in ihrer Studie, dass Kinder in Lernsettings tatsächlich alleine beginnen mit den zur Verfügung stehenden Materialien zu spielen und diese dadurch auch zu erkunden. Erwachsene hingegen führen lediglich die gestellte Aufgabe aus und warten auf die nächste. Sie brauchen Anweisungen und Anleitungen, wie sie weiter verfahren sollen und ihr Lernen findet durch Vorgaben und Anweisungen statt. (Beals & Bers, 2006, S. 356)

2.5 Technik im Kontext der Geschlechter

In diesem Abschnitt wird die Begeisterung von Mädchen und Jungen für Technik, IT, Wissenschaft und Mathematik beleuchtet (siehe Kapitel 2.5.1). In weiterer Folge wird die aktuelle Situation der Frau in der Technikwelt diskutiert (siehe Kapitel 2.5.2), aufgezeigt, welchen Einfluss eine frühzeitige Berührung mit Technik hat (siehe Kapitel 2.5.3) und wie sich das auf das Selbstbewusstsein junger Menschen und die Rollenbilder auswirkt (siehe Kapitel 2.5.4).

2.5.1 Herangehensweisen von Mädchen und Jungen

Brauner, Leonhardt, Ziefle & Schroeder (2010) zeigen auf, dass zwischen Mädchen und Jungen keine Unterschiede im Verständnis von Programmiercode bestehen. Jedoch werden die Unterschiede bei der Entwicklung von Programmen sichtbar. Jungen gelingt im direkten Vergleich eine höhere Programmqualität als Mädchen. (Brauner et al., 2010, S. 66), (Tsan, Boyer & Lynch, 2016, S. 393) Brauner et al. (2010) führen dies auf die niedrigere Selbstwirksamkeitserwartung der Mädchen zurück, die zu Beginn der Studie ihre Computer-Kompetenzen deutlich schwächer einschätzten als die Jungen. (Brauner et al., 2010, S. 66 & S.69) Auch Stoilescu & Beckford (2009) berichten davon, dass Mädchen etwas losgelöster von Regeln an die Programmierung heran gehen, dafür aber ihr Tun immer in Bezug auf ihr persönliches Umfeld und Lernen beziehen. Junge Männer sind eher formaler und folgen mehr den Regeln der Programmierpraxis. Beide Geschlechter programmieren aber gleich motiviert. (Stoilescu & Beckford, 2009, S. 3170)

Weiter führen Brauner et al. (2010) an, dass ihre Untersuchungen ergaben, dass Jungen zwar einen Zusammenhang zwischen Mathematik und Computern sehen, Mädchen hier aber einen klaren, deutlichen sogar engen Zusammenhang erkennen. (Brauner et al., 2010, S. 69) Mädchen schätzten ihre Kompetenzen in Arithmetik, Geometrie und Mathematik allgemein als deutlich schlechter ein, als die Jungen. (Brauner et al., 2010, S. 66) Was die Autorinnen und Autoren jedoch verwundert, denn gemäß Studien aus den USA gibt es keine geschlechtsspezifischen signifikanten Unterschiede bei standardisierten Mathematiktests. (Brauner et al., 2010, S. 69)

Keine geschlechterspezifischen Unterschiede beim Programmieren von einfachen Programmen konnten Pane & Myers (2002, S. 685) feststellen. Alter und Geschlecht haben den Beobachtungen von Hyun & Yoon (2009) zu Folge auch keinen Einfluss auf die Nutzung eines Roboters. (Hyun & Yoon, 2009, S. 680) Beide Geschlechter mögen Robotik und sind sehr motiviert damit zu arbeiten. (Johnson, 2003, S. 21)

Die Evaluierungen der Software Stagecast Creator zeigen auf, dass Kinder (im Alter von 6 bis 17 Jahren), unabhängig von ihrem Geschlecht, in der Lage waren, innerhalb von 15 Minuten erste Programme zu entwickeln. Interessant zu beobachten ist, dass Jungen eher Konflikt-Spiele entwickelten und Mädchen eher Spiele, die auf Kooperationen basieren. (Smith, Cypher & Tesler, 2000, S. 80)

Woods, Davis, Dautenhahn & Schulz (2005) ließen Kinder im Alter von 9 bis 11 Jahren Geschichten mit Robotern zum Thema Mobbing schreiben. Dabei fanden sie heraus, dass beide Geschlechter sich mit Robotern identifizieren konnten und entsprechende Geschichten verfassten. Aufgefallen ist dabei, dass Jungen eher Science-Fiction Geschichten schrieben, die oftmals ein neutrales Ende hatten. Mädchen jedoch legten mehr Wert auf Emotionen und versuchten stets ein Happy-End zu gestalten. (Woods et al., 2005, S. 388)

2.5.2 Frauen in der Technikwelt

Interessant ist die Tatsache, dass es nach Clayton, von Hellens & Nielsen (2009) geografische Unterschiede beim Thema Frauen in der IT gibt. (siehe auch: (Schubert & Schwill, 2011, S. 298 ff.)) Während z.B. in Frankreich, Spanien und anderen Ländern die Frauenquote in der IT konstant hoch ist, ist sie z.B. in Deutschland, Österreich und den skandinavischen Ländern eher niedrig und fallend. (Clayton, von Hellens & Nielsen, 2009, S.

154) Die Autorinnen untersuchten die kulturellen Einflüsse, die für die Entwicklung und Reproduktion von Stereotypen verantwortlich zeichnen und erarbeiteten, dass die Lehre und das Bildungssystem sehr stark zu der Stereotypen-Ausprägung bei den jungen Mädchen beitragen. (Clayton, von Hellens & Nielsen, 2009, S. 156) Auch Birbaumer, Tolar & Wagner (2006) zeigen in ihrer Studie über unterschiedliche Biographien europäischer Frauen in der IT auf, dass sich Frauen eher aus ihrem Kontext heraus für einen IT-Beruf oder eine IT-Tätigkeit entscheiden. (Birbaumer, Tolar & Wagner, 2006, S. 67)

In Nordamerika herrscht ebenso ein Mangel an Frauen in technischen sowie wissenschaftlichen Berufen vor. (Lemons & Parzinger, 2007, S. 91), (Patrick, Mantzicopoulos & Samarapungavan, 2009, S.166) Der Grund dafür liegt laut Lemons & Parzinger (2007) ebenso wie Clayton, von Hellens & Nielsen (2009) berichten, in den Rollenbildern und Geschlechtsschemata⁵, die schon in sehr frühen Entwicklungsstadien beim Menschen geformt und gefestigt werden. Die Vermittlung traditioneller Geschlechtsschemata führt dazu, dass diese später ebenso er- und gelebt werden. Hingegen kann eine Vermittlung nicht traditioneller Geschlechtsschemata bzw. nicht geschlechterspezifischer Verhaltensweisen dazu führen, dass im weiteren Verlauf diese Personen weniger den traditionellen Verhaltensweisen verhaftet agieren. (Lemons & Parzinger, 2007, S. 92) In diesem Zusammenhang spielen das soziale Gefüge und der Androzentrismus eine große Rolle. Hierbei legt die männliche Sichtweise einen Standpunkt fest und alles, was davon abweicht, ist ein Widerspruch zum Standard. (Lemons & Parzinger, 2007, S. 93)

In den 1970er Jahren wurden die Unterschiede in der Gesellschaft bez. der Geschlechter klar hervorgehoben:

„While a man’s success is usually attributed to skill, a woman’s success on the identical task is usually attributed to luck; conversely, man’s failure is attributed to bad luck while woman’s failure is attributed to low ability.“ (Lemons & Parzinger, 2007, S. 93)

Untersuchungen zeigen auf, dass Frauen in der IT weniger traditionellen Geschlechtsschemata folgen und auch Männer in dieser Fachrichtung diesen weniger abgewinnen können. Hingegen zeigen Kontrollgruppen, sowohl bei Männern als auch bei Frauen, die nicht in der IT tätig sind, dass diese eher traditionelleren Geschlechtsschemata folgen. (Lemons & Parzinger, 2007, S. 96)

Solga & Pfahl (2009) setzen sich in ihrer Arbeit mit dem Themenbereich „Gender im technisch-naturwissenschaftlichen Bereich“ auseinander. Die Problematik fassen sie selbst in drei Punkten zusammen: „(1) in den Natur- und Technikwissenschaften ‚verliert‘ man Frauen sehr früh im Lebensverlauf. (2) Wie beim Durchlauf eines Trichters [...] verringert sich mit jeder Bildungs- und Karrierestufe die Anzahl von Frauen in diesen Fächern und Berufen. (3) Die Erhöhung des Interesses für Technikwissenschaften von Mädchen und Frauen ist nicht nur ‚ein Problem‘ für das Bildungssystem, sondern ganz wesentlich auch des Arbeitsmarktes.“ (Solga & Pfahl, 2009, S. 155)

Nach wie vor gibt es eine starke Konzentration von Mädchen in Hauswirtschafts-, Bekleidungs- oder Fremdenverkehrsschulen, sowie Lehrberufen, die traditionell eher von Frauen ausgeübt werden und wirtschaftlichen Berufen. Demgegenüber steht die starke Konzentration von Männern in technischen Ausbildungsrichtungen. (Knoll & Ratzler, 2010, S. 30) Wobei diese Aufteilung nicht biologisch bedingt ist, jedes der beiden Geschlechter kann die jeweiligen Berufe und Ausbildungen ausüben. Es sind vielmehr die gesellschaftlichen Geschlechterzuordnungen, die über Eignung und Nicht-Eignung entscheiden. (Knoll & Ratzler, 2010, S. 37), (Solga & Pfahl, 2009, S. 156) 2008 waren nur rund 20 % der Studierenden in technischen Universitäten und ausgewählten Fachhochschul-Studiengängen

⁵ Geschlechtsschemata sind in Wechselbeziehung stehende Netzwerke mentaler Assoziationen, die Informationen über die Geschlechter repräsentieren. Diese Schemata sind keine passiven Kopien ihrer Umgebung, sondern aktive Konstruktionen, die für Fehler und Verzerrungen anfällig sind. (Ruble, Martin & Berenbaum, 2007 S. 908)

Frauen. (Knoll & Ratzer, 2010, S. 43) Noch kleiner fällt der Anteil der weiblichen Informatikstudierenden z.B. in Kanada aus, wo nur knapp 14 % der Studierenden in Informatik Frauen sind. (Stoilescu & Beckford, 2009, S. 3168) Ähnlich sieht es beim Personal des akademischen Mittelbaus aus, (Knoll & Ratzer, 2010, S. 45) eklatanter ist der Unterschied bei den Professorinnen und Professoren - hier sind nur mehr rund 7 % Frauen. (Knoll & Ratzer, 2010, S. 46)

Pohl & Lanzenberger (2008) heben in Bezug auf weibliche Informatikstudierende zwei interessante Aspekte hervor: Zum einen entscheiden sich Frauen eher kurzfristig (innerhalb von 12 Monaten vor Studienbeginn) Informatik zu studieren und zum anderen beginnen sie dieses Fach auch eher allein zu studieren, ohne einen Freundeskreis, der das gleiche studieren möchte. Im Gegensatz zu den männlichen Studierenden, wo sich ein guter Teil schon langfristig für das Studium entschieden hat und sich auch mit einem bestehenden Freundeskreis für ein Informatikstudium inskribiert. (Pohl & Lanzenberger, 2008, S. 188)

In unserer Gesellschaft existiert noch immer ein zum Teil antiquiertes technikzentriertes Bild von Berufen in der Technik. Außen vor bleiben die in der heutigen Zeit enorm wichtigen „Social Skills“, wie kommunikative oder soziale Kompetenzen. Auch Management- oder Team-Fähigkeiten sind unabdingbar bei Berufen in der Technik. (Knoll & Ratzer, 2010, S. 55)

2.5.3 Frühzeitige Berührung mit Technik

Zahlreiche Forschungsergebnisse belegen, dass Kinder, die schon frühzeitig STEM- (Science Technology Engineering Mathematics) bzw. MINT- (Mathematik Informatik Naturwissenschaften Technologie) Unterricht erfahren durften, deutlich weniger stereotypische, genderbasierte Sichtweisen auf Karrieren im MINT-Bereich inne haben und weniger Hindernisse sehen, im technischen Bereich einen Beruf auszuüben. (Sullivan & Bers, 2015, S. 3), (Sullivan, Kazakoff & Bers, 2013, S. 206)

Eine frühe Vermittlung von Grundlagen der Wissenschaft und Technik erscheinen daher absolut notwendig, um in weiterer Folge auch die Ausprägung von traditionellen Geschlechtsschemata einzuschränken. Denn wie Patrick, Mantzicopoulos & Samarapungavan (2009) feststellen, haben Mädchen und Jungen im Kindergartenalter das gleiche Interesse an Wissenschaft und Technik, wenn man ihnen die Möglichkeiten gibt. Es sind in den frühen Jahren der Entwicklung diesbezüglich noch keine bzw. kaum Unterschiede wahrnehmbar. (Patrick, Mantzicopoulos & Samarapungavan, 2009, S. 182 ff.)

Obwohl es nach Solga & Pfahl (2009) nur wenige Studien gibt, die sich dem Thema „Mädchen und Jungen und deren Beschäftigung mit natur- und technikwissenschaftlichen Phänomenen“ annehmen (Solga & Pfahl, 2009, S. 158), ist es allgemein bekannt, dass „bereits in der frühen Kindheit das Interesse an Naturwissenschaft und Technik von Jungen und Mädchen unterschiedlich gefördert wird.“ (Solga & Pfahl, 2009, S. 159) Somit lässt sich dies als Ursache für ein mögliches Desinteresse des weiblichen Geschlechts an Technik und Naturwissenschaften festmachen. Es zeigt sich, dass Mädchen bereits im Vorschulalter und in weiterer Folge in der Grundschule über geringere Vorerfahrungen im Umgang mit Technik verfügen als ihre männlichen Altersgenossen. Beachtet man, dass die vorschulische Auseinandersetzung mit unterschiedlichen Themen von bedeutender Wichtigkeit ist, erscheint es an dieser Stelle beinahe unmöglich dieses Erfahrungsdefizit wett zu machen. Jedoch zeigen „biografieanalytische Rekonstruktionen [...], dass junge Frauen, die naturwissenschaftlich-technische Studiengänge wählen, auf eine vaterzentrierte Sozialisation zurückblicken.“ (Solga & Pfahl, 2009, S. 159) Bei diesen Frauen spielte der Vater eine wichtige und entscheidende Rolle in der frühkindlichen Erziehung. Väter haben in diesen Fällen „explizit ermutigt, sich mit wissenschaftlichen und technischen Fragestellungen auseinanderzusetzen.“ (Solga & Pfahl, 2009, S. 159)

Weiterführende Analysen zeigen einen deutlichen Zusammenhang zwischen den elterlichen Berufen (wenn diese im naturwissenschaftlichen Bereich angesiedelt sind) und den

naturwissenschaftlichen Kompetenzen der Kinder. (Solga & Pfahl, 2009, S. 159) Auch das generelle Interesse der Eltern für diese Themenbereiche spielt eine wichtige Rolle im Vorschulalter der Kinder. „Bereits im Vorschulalter wirken neben den Eltern auch Erfahrungen mit Erzieherinnen und Erziehern [...] auf die geschlechts(un)typische Entwicklung von Kindern ein. Die Neugier und das forschende Lernen im spielerischen Umgang mit Technik von Kleinkindern werden von diesen zumeist geschlechterstereotyp bewertet und gefördert. Mädchen und Jungen lernen so durch gelebte Vorbilder in der Familie und durch Anregungen und Rückmeldungen im Kindergarten die für ihr jeweiliges Geschlecht und soziales Milieu passenden Verhaltensweisen kennen.“ (Solga & Pfahl, 2009, S. 160)

Auch wenn sich Papert (1993) nicht unbedingt dem Thema Gender gewidmet hat, so steckt in seinem Werk „Mindstorms“ ein möglicher Ansatz zur Vermeidung der Geschlechterzuordnungen:

„The understanding of learning must be genetic. It must refer to the genesis of knowledge. What an individual can learn, and how he learns it, depends on what models he has available. This raises, recursively, the question of how he learned these models.“ (Papert, 1993, S. xix)

Wentzel (2008) stellt Rollenvorbilder vor und zeigt auf, dass junge Frauen, die in technischen oder informationstechnischen Berufen arbeiten, bereits in ihrer Kindheit mit diesen Themen in Berührung gekommen sind. Ein Umfeld, welches die Auseinandersetzung mit Technik und Wissenschaft ermöglicht, kann nachhaltig das Interesse und die Beschäftigung mit diesen Themen fördern. (Wentzel, 2008, S. 10) Junge Frauen differenzieren dabei später aber nicht, ob es sich bei den Tätigkeiten und der Beschäftigung mit Technik und Wissenschaft um „männertypische“ Themen handelte. (Wentzel, 2008, S. 28) Die Schule, im Speziellen der Fachunterricht, spielt bei der Berufsorientierung eine wichtige Rolle, jedoch verstärkt die Schule oftmals Geschlechterklischees mehr, als dass diese durchbrochen werden. Das gipfelt sogar darin, dass Mädchen technische Berufe von Lehrkräften ausgeredet werden. (Wentzel, 2008, S.42) Die von Wentzel (2008) befragten jungen Frauen betrachten ihre technischen Berufe als „interessant, faszinierend und teilweise auch kreativ.“ (Wentzel, 2008, S. 28) Bei der Berufswahl der befragten jungen Frauen spielte die intrinsische Motivation eine sehr hohe Rolle, wobei die extrinsische Motivation hauptsächlich aus der Richtung der Arbeitsmarktchancen auszumachen war. (Wentzel, 2008, S. 29)

2.5.4 Selbstbewusstsein und Rollenbilder

Bischof (2010) beobachtete in ihrer Forschungsarbeit, dass Mädchen und Jungen ein sehr ähnliches Interesse an Informatik und Technik aufweisen. Auch glauben beide Geschlechter, dass beide Geschlechter „gleich gut“ für Informatik geeignet sind. Darüber hinaus verfügen die befragten Volksschülerinnen und -schüler über noch keine ausgeprägten Genderklischees. (Bischof, 2010, S. 177) Bezüglich der Eignung und Leistungen konnten beim internationalen Informatikwettbewerb für 5- bis 12-jährige Kinder, Bebras⁶, keine signifikanten Unterschiede in Informatik zwischen Jungen und Mädchen festgestellt werden. (Dagiene & Futschek, 2010, S. 10) Ähnliche Beobachtungen, auch in Bezug auf Wissenschaftsbildung, berichten Saçkes, Trundle, Bell & O’Connell (2010, S 229). Gibbs & Roberts (2003) bestätigen ebenfalls, dass junge Mädchen Computer noch nicht als „mehr für Männer geeignet“ einstufen. (Gibbs & Roberts, 2003, S. 41) Dies kann durch McKenney & Voogt (2010) belegt werden, die in einer Umfrage die Selbsteinschätzung von jungen Kindern zu Computerkenntnissen und Nutzung von Computern erhoben haben. Hierbei

⁶ <http://www.bebas.org/>, (23.04.2018). Der Bebras-Wettbewerb hat zum Ziel, 5- bis 12-jährige Kinder für Informatikkonzepte zu begeistern. Die Grundidee ist es, interessante Probleme aus dem Bereich der Informatik zu stellen und lösen zu lassen. Durch das Lösen der Aufgaben lernen die Teilnehmerinnen und Teilnehmer wiederum Neues zu den jeweiligen Konzepten. (Dagiene & Futschek, 2010, S. 1) In Österreich heißt der Wettbewerb: Biber der Informatik. <http://wettbewerb.biber.ocg.at/>, 23.04.2018

konnten keine Unterschiede zwischen den Geschlechtern festgestellt werden. (McKenney & Voogt, 2010, S. 663)

„Es geht also darum, Professionelle zu befähigen, eine Geschlechter-De-Konstruktion durch Doing Gender-Analysen in der Pädagogik der Kindheit umzusetzen und sie gemeinsam mit allen Beteiligten analytisch zu begründen und interaktiv umzusetzen.“ (Giebeler, 2010, S. 36)

Wilgosh (2002) hebt hervor, dass es schon in den frühen Jahren der Entwicklung von Kindern professioneller Unterstützung bedarf, damit Mädchen mit dem Druck der männerdominierten Gesellschaft umgehen können und Jungen ein Platz zur Verfügung gestellt werden kann, wo sie verletzlich sein dürfen. Um die persönliche Weiterentwicklung von jungen Menschen zu unterstützen empfiehlt Wilgosh (2002) persönliche, gleichgeschlechtliche Mentorinnen und Mentoren, sowie Rollenvorbilder. (Wilgosh, 2002, S. 255)

3 Informatikdidaktik und Computational Thinking

Die Informatikdidaktik als Querschnittsthema zwischen Informatik als Fachdisziplin, Didaktik allgemein, Pädagogik und Psychologie (Schubert & Schwill, 2011, S. 14) gilt es genauer zu betrachten. Daher diskutiert dieses Kapitel, was unter dem Begriff der Informatik im Rahmen der gegenständlichen Forschungsarbeit zu verstehen ist und welche Werte sowie Konzepte mit ihrer Hilfe, im Rahmen von Informatik-Unterricht, vermittelt werden können. (Kapitel 3.1)

Da es in Österreich keinen curricularen Informatikunterricht im Kindergarten für Kinder im Alter von 3 bis 6 Jahren gibt, zieht die Verfasserin der gegenständlichen Arbeit den Informatikunterricht in der Schule für Überlegungen, Betrachtungen und Beobachtungen heran, wie Informatik-Unterricht für diese Zielgruppe adäquat gestaltet werden könnte.

Anschließend diskutiert die Autorin die Kompetenzentwicklung und die Förderung von Kompetenzen, die mit dem Einsatz von IKT als Unterrichtsfach und als Hilfsmittel für junge Kinder einhergehen. (Kapitel 3.2) Als zentrale Kompetenz oder wesentlicher Kern der Aktivitäten sind hierzu das „Higher-Order Thinking“ und das „Computational Thinking“ anzuführen. (Kapitel 3.3)

3.1 Didaktische Überlegungen zum Praxiseinsatz

Auch wenn IKT zunehmend Einzug in den Alltag von uns Menschen hält, so lässt das fachliche, reflektierte Wissen rund um IKT oftmals jedoch zu wünschen übrig. Es mangelt nicht nur an IKT-Kenntnissen, sondern vor allem an Informatikkenntnissen. (Döbeli, 2010, S. 37)

„Die Verbreitung von Standardsoftware in den [19, Anm. d. Verf.]80er-Jahren hat scheinbar Programmierkenntnisse überflüssig gemacht und schließlich zur Abschaffung des Informatikunterrichts in Schweizer Gymnasien geführt. Bereits sind erste Stimmen zu hören, heutige Hard- und Software sei so einfach geworden, dass auch keine ICT-Anwendungskenntnisse mehr vermittelt werden müssten und somit auch das Thema ICT in der Schule überflüssig sei.“ (Döbeli, 2010, S. 38)

Die Allgegenwärtigkeit von Informationstechnologie sowie Informations- und Kommunikationssystemen in unserer Gesellschaft, im beruflichen Umfeld, sowie in der Forschung und Wissenschaft zeigt auf, dass eine Auseinandersetzung mit diesem Thema in der Schule unumgänglich erscheint. Die schulische Informatikbildung setzt sich hierbei heute vor allem mit drei Schwerpunkten auseinander:

1. Einsatz von Informatiksystemen als Unterstützung von Lernvorgängen
2. Erlernen der Bedienung von konkreten Systemen
3. Vermittlung allgemeiner und langlebiger Grundlagen der Informatik

(Hubwieser, 2007, S. 43 f.)

Die Synthese aus diesen drei Schwerpunkten bezeichnet Hubwieser (2007) als informatische Bildung. Für einen sinnvollen Einsatz von Informatiksystemen zur Unterstützung von Lernvorgängen ist eine grundlegende Schulung in der Bedienung unumgänglich. Das erfordert auch Kenntnis der zu Grunde liegenden Basiskonzepte, um eine effiziente Nutzung des Systems zu ermöglichen. (Hubwieser, 2007, S. 48)

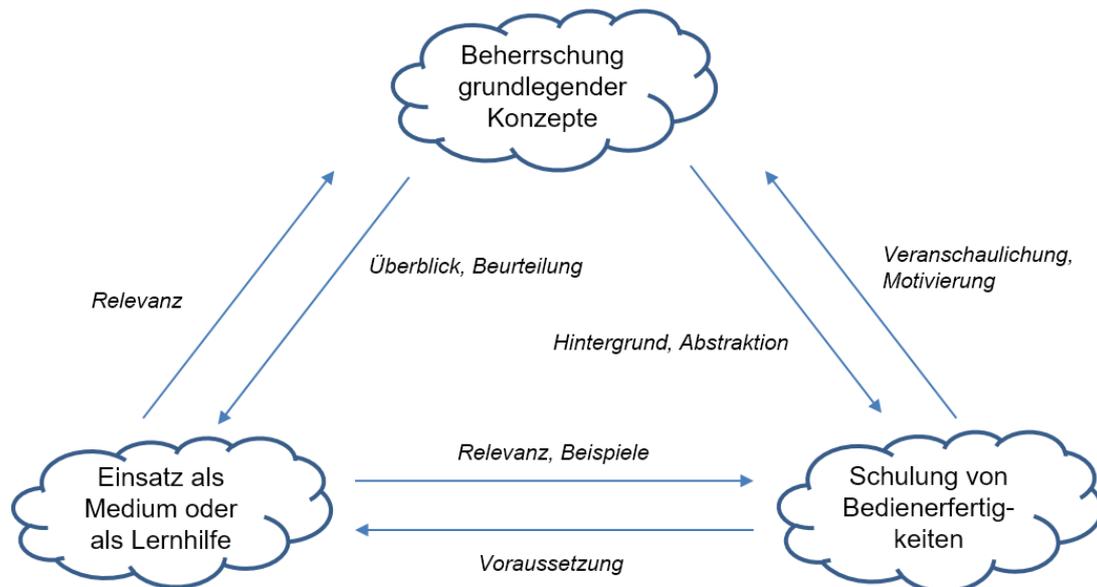


Abbildung 3: Informatische Bildung, eigene Darstellung nach (Hubwieser, 2007, S. 49)

Damit die erlernte Bedienung eines Systems auf die von ähnlichen Systemen übertragen werden kann, bedarf es hierbei ebenfalls der Kenntnis der Basiskonzepte. Jedoch ist zur Vermittlung der Basiskonzepte der Informatik ein Einsatz von Informatiksystemen unerlässlich, wenn man nicht auf einer rein theoretischen und abstrakten Basis verweilen möchte. Dafür sind aber wiederum Bedienerfertigkeiten notwendig, um Informatiksysteme zur Unterstützung von Lernvorgängen effektiv und effizient einsetzen zu können. (Hubwieser, 2007, S. 48) Diese gegenseitigen Abhängigkeiten sind in Abbildung 3 dargestellt. Ähnliche Abhängigkeiten im Unterricht der Informations- und Kommunikationstechnologien sind bei Hartmann, Näf & Reichert (2006, S. 4) zu finden.

Folgend der Begründung der Notwendigkeit von Informatik als Unterrichtsfach sowie der Darlegung des Beitrags von Informatik zur Allgemeinbildung nach Hubwieser (2007) zeigt sich, dass die zunehmende Informationsflut, der Arbeitsmarkt, (Hubwieser, 2007, S. 58) die wirtschaftliche Bedeutung und Allgegenwart der Informatiksysteme (Hubwieser, 2007, S. 59) fordern, dass junge Menschen sich in der heutigen Informationsgesellschaft zurechtfinden können. Auch die Kompetenzen zur Bewertung von Informationen und Strukturierung selbiger, im Besonderen durch eine effiziente Nutzung von Programmen, werden als notwendig erachtet. (Hubwieser, 2007, S. 58)

Eine große Herausforderung stellt, in Anbetracht der Vielseitigkeit der Informatik, die Gestaltung des Einstiegs- oder Anfangsunterrichts dar. Insbesondere die Heterogenität der Gruppe in Bezug auf ihr Vorwissen zum Thema Informatik und IKT stellt eine Hürde dar, die es zu meistern gilt. (Schubert & Schwill, 2011, S. 287) Hierzu bieten sich unterschiedliche Zugänge in der Gestaltung des ersten Unterrichts an. Neben dem programmiersprachlichen Zugang (Schubert & Schwill, 2011, S. 289 ff.), der allgemein hin als „klassische Informatik“ bezeichnet wird, werden auch der systemanalytische Zugang (Schubert & Schwill, 2011, S. 291 ff.), der Zugang über Lernumgebungen (Schubert & Schwill, 2011, S. 293 ff.) sowie der projektorientierte fächerübergreifende Zugang (Schubert & Schwill, 2011, S. 297 ff.) stets interessanter und in den Fokus gerückt. Eine Zielgruppenanalyse und eine klare Definition von zu erwartenden Lernergebnissen sind notwendig, um eine geeignete Auswahl an Methoden für den Start in das Unterrichtsfach Informatik treffen zu können.

In der Slowakei wurde 2008 Informatik als Pflichtgegenstand in den Volksschulen ab der zweiten Schulstufe eingeführt. Der Lehrplan umfasst die fünf Themenbereiche: Informationsverarbeitung (Texte, Grafiken, Multimedia und Datenstrukturen); Kommunikation mit IKT (Web, E-Mail, Chat, ...); Methoden, Problemlösen und algorithmisches Denken (Programme entwerfen, Agenten programmieren, Zusammenhänge erkennen, ...);

technische Grundlagen der IKT (Hardware, Betriebssysteme, ...) und Informationsgesellschaft (Risiken, Privatsphäre, gesellschaftliche Auswirkungen, ...). (Kabátová, Kalaš & Tomcsányiová, 2016, S. 129)

3.1.1 Informatik – Was ist das?

In der Literatur wird Informatik als Kunstwort definiert, das sich aus den beiden Wörtern Information und Automatik zusammensetzt, welches weiters keinen Wortstamm des Wortes Computer enthält und somit auch nicht zwangsläufig den Computer als zentrales Element beinhalten müsste. (Mittermeir, 2010, S. 72), (Federer, Futschek & Rugg, 2009, S. 19) Für Kabátová, Kalaš & Tomcsányiová (2016) stellt Informatik zum einen das breite wissenschaftliche Feld hinter digitalen Technologien und zum anderen einen Oberbegriff für alle Konzepte dar, die mit digitaler Technologie, Information oder den korrespondierenden Hintergrundtheorien zu tun haben. (Kabátová, Kalaš & Tomcsányiová, 2016, 128)

Als „digital literacy“ definieren Kabátová, Kalaš & Tomcsányiová (2016) ein Set an Basisfähigkeiten, die jeder während seiner Ausbildung erwerben sollte, um digitale Technologien (nicht nur Computer, sondern alle digitalen Geräte) effektiv, sicher und zielführend zur Lösung von Alltagsproblemen einsetzen zu können. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 129)

Im Fokus der Informatikausbildung steht einerseits die Fähigkeit, den Computer vertraut und effektiv zu benutzen. Zum anderen sollten Kompetenzen vermittelt werden, die die Basiskonzepte der Informatik (siehe Kapitel 3.1.3), wie etwa Algorithmen oder Datenrepräsentation, beinhalten. (Antonitsch, Gigacher, Hanisch & Sabitzer, 2013, S. 118)

Wird zum einen „digital literacy“ bzw. „computer literate“ oftmals mit der Bedienung eines Computers auf Niveau des Europäischen Computerführerscheins (ECDL) gleichgesetzt (Mittermeir, Bischof & Hodnigg, 2010, S. 144), sieht Antonitsch (2013) immer noch einige Missverständnisse in Bezug auf die öffentliche Wahrnehmung von Informatik. Dazu nennt Antonitsch (2013) einige Beispiele aus Gesprächen zwischen Eltern und Lehrerinnen und Lehrern bzw. zwischen Lehrenden und Schülerinnen und Schülern. Eine verbreitete Sichtweise ist die, bei der jegliche Computernutzung mit informatischem Denken oder Handeln verwechselt wird. Ein anderes Missverständnis ist die Annahme, in einer von anderen erstellten virtuellen Welt aus einer Menge vorgegebener Menüpunkte und Aktionen zu wählen und damit ein vorgegebenes Ziel zu erreichen sei Informatisches Handeln. Antonitsch (2013) führt diese Auffassungen, die seinen Beobachtungen nach in der Sekundarstufe sichtbar werden, auf die Art und Weise zurück, wie Kinder in jüngeren Jahren mit virtuellen Welten und IKT in Berührung kommen – erfolgt dies eher passiv durch Medienkonsum, sind die Kinder in späteren Jahren daran gewöhnt und in Folge nicht bereit, allzu viel in eigenständige Problemlösungen zu investieren. (Antonitsch, 2013, S. 360 f.)

In Österreich wird das Ziel des Informatikunterrichts vielerorts mit „computer literate“ in Verbindung gebracht, d.h. allzu oft wird das Curriculum des ECDL als „Informatik“ unterrichtet und keine informatischen Konzepte, was der Wissenschaftsdisziplin nicht gerade positiv zuträgt. (Mittermeir, Bischof & Hodnigg, 2010, S. 145) Die Auseinandersetzung mit Informatik fördert jedoch wesentliche Kompetenzen: Die Kreativität durch das Aufzeigen unterschiedlicher Wege zur Lösung von Problemen, die Konstruktions-Fähigkeiten durch algorithmisches Denken und die Erstellung von (virtuellen) Artefakten sowie Komplexitätsbeherrschung. (Antonitsch, 2013, S. 362) Siehe dazu auch Kapitel 3.1.4.

Das Lernen von IKT und Informatik (wie auch von Naturwissenschaften, siehe u.a. (Streim et al., 1995)) soll durch Erfahrung geschehen, so dass Unterrichtsszenarien idealerweise mit Fokus auf aktives Lernen, Kreativität und Erforschung gestaltet werden. Hierbei soll bewusst auf den Computer als Lehrmaterial verzichtet und der Unterricht zunehmend auf die Konzepte fokussiert werden. (Antonitsch et al., 2013, S. 119) Neben der interdisziplinären Verankerung von Informatik in den Unterrichtsalltag sieht Döbeli (2010) die große Chance in der Vermittlung von informatischen Konzepten, wenn diese angreifbar und erlebbar gemacht

werden. (Döbeli, 2010, S. 42) Eines dieser Konzepte ist z.B. „Computer Science unplugged“⁷. (Wohl, Porter & Clinch, 2015, S. 1 f.)

Lernende sollen den Computer somit als Werkzeug zur Lösung von Problemen wahrnehmen oder zur Unterstützung von Problemlösungsprozessen heranziehen. (Antonitsch et al., 2013, S. 119), (Freudenberg, 2009, S. 88 ff.) Jedoch haben genau damit sehr viele Lernende ein Problem – sie verstehen oftmals die Problemstellung nicht:

„... an increasing number of learners fails to understand written problem descriptions and is unable to enter a step-by-step problem solving process, particularly when the computer is switched on. Learners tend to ask for ‘a button to solve the problem instantly’ instead.“ (Antonitsch et al., 2013, S. 119)

Eine der größten Herausforderungen, die es zu meistern gilt, um diesem Problem entgegenzuwirken, besteht wohl darin, dass in vielen Bildungseinrichtungen nicht ausreichend Geräte vorhanden sind, um allen Kindern gleichzeitig die Möglichkeit zu bieten ihre eigenen Erfahrungen zu machen. Gerade auch deswegen erscheint es sinnvoll, im Unterricht den Fokus mehr auf die Konzepte der Informatik zu legen und weniger auf die Technik an sich. (Antonitsch et al., 2013, S. 120), (Mittermeir, Bischof & Hodnigg, 2010, S. 147 ff.)

3.1.2 Bildungswerte der Informatik

Die Didaktik der Informatik ist eingebettet in „einen Kreis unterschiedlicher Wissenschaften und Institutionen sowie eine typische Fragestellung, die in dem jeweiligen Bereich bearbeitet wird und [...] [deren, Anm. d. Verf.] zugehörige Antworten die Didaktik der Informatik zum Aufbau ihrer Lehre verwendet.“ (Schubert & Schwill, 2011, S. 14) Hierzu führen Schubert & Schwill (2011) die Psychologie, die Pädagogik, die Schule und die Informatik selbst an. (Schubert & Schwill, 2011, S. 14)

Die Informatikbildung bereitet junge Menschen auf zukünftige Lebenssituationen vor (Hubwieser, 2007, S. 62), unterstützt bei der Bildung eines Weltbilds, versucht Barrieren in der Nutzung (insbesondere gesellschaftliche Hürden) abzubauen, bietet eine Anleitung zum „kritischen Vernunftsgebrauch“ von Informatiksystemen (Hubwieser, 2007, S. 63) und stärkt das Lernenden-Ich. (Hubwieser, 2007, S. 64)

Nach Futschek (2003) vertritt die Informatik ihre eigenen, beständigen und nachhaltigen Bildungswerte:

- „Bewältigung von Systemen großer Komplexität
- Modellbildung und Abstraktion
- Kooperatives und interdisziplinäres Arbeiten
- Algorithmisches Denken
- Kreatives Gestalten und gesellschaftlich relevante Automatisierung“

(Futschek, 2003, S. 380)

Die Informatikbildung dient ebenso zur Berufs- und Studiumsvorbereitung, da sie Methoden und Kenntnisse „zur Beschreibung und Lösung von Problemen, zur Beschaffung und Darstellung von Informationen, zur Beherrschung und effizienten Nutzung von Informationssystemen“ beiträgt. (Hubwieser, 2007, S. 65) Es geht weniger darum, dass junge Menschen gleich Programmiererinnen und Programmierer werden, vielmehr sollen die zugrundeliegenden Konzepte vermittelt werden. (Reichert, Nievergelt & Hartmann, 2005, S. 9)

⁷ www.csunplugged.org, 23.04.2018

Es herrscht Einigkeit darüber, dass mit dem Unterricht in „digital literacy“ in jungen Jahren begonnen werden soll. Grundlagen wie Dokumente erstellen, Präsentationen und Zeichnen oder über digitale Medien miteinander kommunizieren sollen mit 12 Jahren vorhanden sein. Diese Einigkeit gilt jedoch noch nicht für die Informatikausbildung. Das CSTA (Computer Science Teacher Association) Framework für K-12⁸ Informatik Standards⁹ ist ein gutes Beispiel für Bemühungen einen Konsens zu finden: Das Framework stellt ein umfassendes Set an Lernzielen zur Verfügung und bezeichnet fünf Stränge, die das Lernen von Informatik durch alle Altersgruppen leiten sollen. (Antonitsch et al., 2013, S. 118)

Neben „collaboration“ (Kollaboration), „computing practice and programming“ (Computereinsatz und Programmieren), „computers and communications devices“ (Computer und Kommunikationsgeräte) und „community, global and ethical impacts“ (gemeinschaftliche, globale und ethische Auswirkungen) ist „Computational Thinking“ einer dieser fünf Stränge (Antonitsch, 2013, S. 362), wobei „Computational Thinking“ als Schlüsselkompetenz des 21. Jahrhunderts hervorgehoben wird. Das letzte Kindergartenjahr und die österreichische Volksschule können zum Level 1 des Frameworks zugeordnet werden. (Antonitsch et al., 2013, S. 118)

Dem Gegenstand der vorliegenden Arbeit entsprechend, werden in Anlehnung an Antonitsch (2013) die Teile des Frameworks genauer vorgestellt, die das letzte Kindergartenjahr miteinschließen. Kinder sollten nach dem Kindergarten und den ersten drei Schuljahren in der Lage sein,

1. Computational Thinking
 - a. Technologie zu nutzen, um altersgerechte Probleme zu lösen.
 - b. Schreibprogramme, Digitalkameras und Zeichenprogramme zu verwenden, um eigene Gedanken, Ideen und Geschichten Schritt für Schritt darstellen zu können.
 - c. zu verstehen, wie Information in einer sinnvollen Reihenfolge sortiert werden kann, ohne den Computer zu verwenden.
 - d. zu erkennen, dass Software entwickelt wird um den Computer zu steuern.
 - e. zu zeigen, wie mit Null und Eins Information repräsentiert werden kann.
2. Kollaboration
 - a. mit Unterstützung von Dritten Informationen zu sammeln und elektronisch zu kommunizieren.
 - b. unter Technologieverwendung kooperativ und kollaborativ mit Lehrpersonen und Peers zusammenzuarbeiten.
3. Computereinsatz und Programmieren
 - a. Technologie einzusetzen um altersgerechte Forschung zu betreiben.
 - b. beim Lernen entwicklungsgerechte Multimediaressourcen als Unterstützung einzusetzen.
 - c. entwicklungsgerechte Multimediaprodukte mit Unterstützung von Dritten zu erstellen.
 - d. ein Set von Befehlen zu entwickeln, das zur Erfüllung einer einfachen Aufgabe genutzt werden kann (z.B. einfache Befehle für Bodenroboter)
 - e. Berufe zu identifizieren, die Informatik und Technologie einsetzen.

⁸ Dies umfasst in den USA die Bildungsspanne vom Kindergarten bis zur 12. Schulstufe, wobei in den USA der Kindergarten im Alter von 5 Jahren beginnt, die Schule mit 6 Jahren. Bildungsangebote, die Kinder vor dem 5. Lebensjahr betreuen werden als Pre-K bezeichnet.

⁹ <http://www.csteachers.org/page/standards>, 23.04.2018

- f. Informationen unter Einsatz von Concept-Mapping Werkzeugen zu sammeln und zu organisieren.
- 4. Computer und Kommunikationsgeräte
 - a. Input- und Outputgeräte zu verwenden um erfolgreich mit Computern und verwandten Technologien zu arbeiten.
- 5. gemeinschaftliche, globale und ethische Auswirkungen
 - a. bei der Verwendung von Technologie und Software als digitale Bürgerin, als digitaler Bürger verantwortungsbewusst zu handeln
 - b. positives und negatives soziales und ethisches Verhalten bei der Nutzung von Technologie zu erkennen.

(CSTA, 2011, S. 13 ff.)

Zurzeit sind zwei Strömungen auszumachen, die unterschiedlich argumentieren, wie Computerkenntnisse und Technologien zu betrachten sind. Die eine Strömung steht dafür ein, dass Kinder über Computerkenntnisse zur Bedienung von Computern verfügen müssen, um besser lernen zu können, produktiver zu werden und mehr Leistung zu erbringen. Die andere Strömung betrachtet mehr die Technologie und Konzepte dahinter und steht dafür ein, dass Kinder lernen sollen, Technologien für sich kreativ und persönlich zu nutzen. (Bers, 2007, S. 6)

Resnick (2014) beschäftigt sich mit der Frage, wie Kinder kreativ lernen können. Er und seine Forschungsgruppe entwickelten ein Konzept, um Kindern dabei zu helfen, als kreative Denkerinnen und Denker aufzuwachsen. Dieser Ansatz basiert auf vier am Konstruktivismus ausgerichteten (und von ihm inspirierten) Kernelementen, die manchmal als „the four P’s of Creative Learning“, die vier Ps kreativen Lernens bezeichnet werden:

1. Projekte: Menschen lernen am besten, wenn sie aktiv an bedeutungsvollen Projekten arbeiten – neue Ideen entwickeln, Prototypen designen und diese iterativ verbessern.
2. Peers: Lernen gedeiht als eine soziale Aktivität, bei der Menschen ihre Ideen teilen, an Projekten zusammenarbeiten und auf der Arbeit anderer aufbauen.
3. Passion (Leidenschaft): Wenn Menschen an Projekten arbeiten, dann kümmern sie sich um diese, arbeiten länger und härter, trotz Herausforderungen und lernen in dem Prozess mehr.
4. Play (Spiel): Lernen umfasst spielerisches Experimentieren – das Ausprobieren neuer Dinge, das Herumbasteln mit Materialien, Grenzen austesten, Risiken eingehen und die fortlaufende Iteration.

(Resnick, 2014, S. 1)

Die „Positive Technological Development“ (PTD)-Sichtweise stellt die Frage, wie Kinder Technologien positiv nutzen können, um sich selbst und der Welt zu helfen. Entwicklungswissenschaftlerinnen und -wissenschaftler betrachten hierzu die kognitiven, persönlichen, sozialen, emotionalen und bürgerlichen Charakteristika junger Menschen. Diese Charakteristika werden zum „sechs C-Modell“ (competence, connection, character, confidence, caring, contribution) zusammengefasst, welches von Bers (2007, 2008) im „C’s by six C’s model“ (competence – content creation, connection – collaboration, character – conduct, confidence – creativity, caring – communication, contribution – community building) erweitert wurde. Details dazu sind bei Bers (2007) und Bers (2008) nachzulesen.

3.1.3 Basiskonzepte der Informatik

Die informelle Beschreibung von Rechenverfahren wurde bereits in der Antike eingesetzt und Algorithmen sind somit eine der ältesten Formen zur Modellierung von Abläufen. (Hubwieser & Aiglstorfer, 2004, S. 19) Ein Algorithmus wird hierzu wie folgt definiert:

„Ein Algorithmus ist ein Verfahren mit einer präzisen (d.h. in einer genau festgelegten Sprache abgefassten) endlichen Beschreibung unter Verwendung effektiver (d.h. tatsächlich

ausführbarer) elementarer (Verarbeitungs-) Schritte.“ (Hubwieser & Aiglstorfer, 2004, S. 19 nach Broy, 1998)

Algorithmen bestehen aus vier Strukturelementen, die zur Darstellung beschreibbarer Berechnungen eingesetzt werden. Zu diesen grundlegenden „Bausteinen“ von Algorithmen zählen:

- Elementare Verarbeitungsschritte, die als unteilbare Verarbeitungsschritte unbedingt ausgeführt werden müssen,
- Sequenzen, die aus hintereinander auszuführenden elementaren Verarbeitungsschritten bestehen,
- Bedingte Verarbeitungsschritte, die nur unter bestimmten Bedingungen ausgeführt werden und
- Wiederholungen, eine Aneinanderreihung von Verarbeitungsschritten wird eine bestimmte Anzahl mal wiederholt, wobei darauf zu achten ist, dass es eine vorgegebene Anzahl an Wiederholungen geben kann oder Wiederholungen, die durch Anfangs- oder End-Bedingungen gesteuert werden.

(Hubwieser & Aiglstorfer, 2004, S. 22 ff.)

Algorithmen werden, wenn sie nicht in maschinell verarbeitbarer Form benötigt werden oder zum Zeitpunkt des Entwurfs, in Form von Pseudocode oder Struktogrammen repräsentiert. (Hubwieser & Aiglstorfer, 2004, S. 27 ff.) Darüber hinaus verfügen gute Algorithmen in der Regel (wenn sie dem E-V-A Prinzip – Eingabe, Verarbeitung und Ausgabe – folgen) über folgende drei Eigenschaften, die ursprünglich mit Turing-Maschinen oder endlichen Automaten in Verbindung gebracht wurden:

- „terminierend, wenn seine Ausführung für jede mögliche Eingabe nach einer endlichen Anzahl von Schritten endet,
- deterministisch, wenn jede Eingabe für jede seiner auszuführenden Anweisungen jeweils eindeutig die Folgeanweisung festlegt (mit Ausnahme der letzten Anweisung, nach der er endet),
- determiniert, wenn er für eine bestimmte Eingabe bei allen Abläufen immer dieselbe Ausgabe liefert.“

(Hubwieser & Aiglstorfer, 2004, S. 25)

Ein Programm verfügt über eine Grammatik, die die Syntax der Programmiersprache vorgibt. Die Wirkung eines Programms, in Form der Semantik, und die Dynamik, als Ergebnis der Reihenfolge des Ablaufs, runden die charakteristischen Merkmale eines Programms ab. (Futschek, 2003, S. 382) Um Programme auf einem Computer ausführbar zu machen, benötigt man einen Compiler oder Interpreter. Während der Compiler zuerst den gesamten Programmtext in Maschinencode übersetzt und anschließend zur Ausführung bringt, übersetzt der Interpreter jeden einzelnen Programmbefehl und führt in direkt aus. (Berns & Schmidt, 2010, S. 32)

Zusammengefasst lässt sich die Erstellung eines Programms nach Hubwieser & Aiglstorfer (2004) in folgende fünf Schritte aufteilen:

1. Menschliche Erfahrungswelt: Erkennen der Aufgabenstellung aus der realen Welt.
2. Modellebene: Starke Vereinfachung der menschlichen Erfahrungswelt und Abstrahierung sowie Abgrenzung zur Lösung der Aufgabenstellung; die Eigenschaften der Aufgabenstellung werden modellhaft wiedergegeben (modelliert).
3. Programmtext: Der Programmtext wird gemäß dem entwickelten Modell in einer höheren Programmiersprache formuliert und geschrieben (Implementierung).
4. Maschinencode: Der erstellte Programmtext wird in Maschinencode übersetzt und direkt ausführbar gemacht.
5. Prozess: Die Ausführung des Maschinencodes lässt einen realen Prozess entstehen.

(Hubwieser & Aiglstorfer, 2004, S. 40)

Zur Umsetzung von Programmen benötigt man darüber hinaus sog. Variablen und Kontrollstrukturen. (Hubwieser & Aiglstorfer, 2004, S. 44 f. sowie S. 55 f.)

3.1.4 Algorithmisches Denken

Das algorithmische Denken wird von zahlreichen anderen kognitiven Fähigkeiten des Menschen beeinflusst: Dem abstrakten, logischen und strukturierten Denken sowie den Kreativitäts- und Problemlösungskompetenzen. (Futschek & Moschitz, 2010, S. 2) Die Kreativität spielt eine gewichtige Rolle, denn je komplexer Problemstellungen sind, desto mehr Freiräume gibt es, um adäquate, individuelle Lösungen zu finden. (Futschek, 2006, S. 160)

Als einen wesentlichen Eckpfeiler ihrer didaktischen Überlegungen für das algorithmische Denken durch Spiel sehen Futschek & Moschitz (2010) die Reflexion zwischen den Lernenden und Lehrenden am Ende jeder Übung. Denn nur so kann eine Verbesserung des gefundenen Algorithmus unterstützt und ein vertiefendes Verständnis für den gefundenen Algorithmus ermöglicht werden. (Futschek & Moschitz, 2010, S. 4 f.) Die Lernenden sind in der Lage durch das spielerische Erleben von algorithmischen Aufgaben aus dem täglichen Leben auch fortgeschrittene Konzepte wie Gleichzeitigkeit, Synchronisierung, Verteilung von Nachrichten, Variablen und vieles mehr zu erlernen. (Futschek & Moschitz, 2010, S. 9) Korrespondierende Erfolge in der Praxis zeigt u.a. Mittermeir (2013) auf.

Da für Kindergartenkinder zum algorithmischen Denken kaum Forschung existiert, betrachtet die Verfasserin der gegenständlichen Arbeit exemplarisch auch Aktivitäten, die mit etwas älteren Kindern durchgeführt wurden.

Forišek & Steinová (2010) stellen einige didaktische Spiele vor, die sich für Jugendliche ab 12 Jahren dazu eignen, den Lernenden in der Schule die Grundlagen und Konzepte der Informatik zu vermitteln. Dazu werden wiederum keine Computer eingesetzt, lediglich Papier und Stift, da diese die zu vermittelnden Inhalte „begreifbarer“ machen. (Forišek & Steinová, 2010, S. 86 f.) Jedes Spiel behandelt mehrere Aspekte der Informationstheorie, z.B. Datencodierung und Datenkomprimierung oder Redundanz der Sprache und Fehlerkorrekturen. (Forišek & Steinová, 2010, S. 91 ff.) Der didaktische Aufbau der Spiele ermöglicht einen tieferen Einblick in das Thema Informatik. (Forišek & Steinová, 2010, S. 97) Auch Weigend (2009) stellt einige Spiele zu den Grundlagen der Informatik vor.

Einen vertiefenden Einblick in die Vermittlung der Informatik mit Hilfe von adäquaten Lehrbüchern für 8- bis 12-Jährige bieten Kalaš & Winczer (2008), die Informatik als einen wesentlichen Teil der modernen konstruktivistischen Lehre ansehen. (Kalaš & Winczer, 2008, S. 229 & S. 239)

Die Unterschiede zwischen Instruktionen und Werten bzw. Variablen und deren Zusammenhänge vermittelt Addison (1997) Kindern ab dem Kindergartenalter mit einer Art Kartenspiel in unterschiedlichen Schwierigkeitsstufen. (Addison, 1997, S. 30 f.) Dabei simulieren die Kinder im Spiel die Funktionsweise eines Computers und lernen, wie der Computer Informationen verarbeitet. Darüber hinaus wird ihnen aufgezeigt wie für die Verarbeitung Probleme in einzelne Schritte zerlegt werden müssen. (Addison, 1997, S. 31) Ähnliches bietet Valente (2004) mit den sog. Computational Cards an, die auf formalen Regeln basieren und stets über einen Input und einen Output verfügen. (Valente, 2004, S. 1 ff.)

3.2 Kompetenzentwicklung durch Einsatz von IKT und Robotern

Der Einsatz von neuen Technologien im Unterricht und in der Lehre, bei denen physische Aktivitäten eine neue Dimension des Lernens hinzufügen, führt zu einer guten Verbindung von theoretischem Wissen und praktischer Anwendung. (Fernaesus & Tholander, 2006, S.

1027) Die Beziehung zwischen Lernenden und Lehrenden wurde durch Technologie um eine Komplexitätsebene, die es zu meistern gilt, erweitert. (Leung, 2006, S. 1)

Eguchi (2007) fasst eine Auflistung an Kompetenzen, die man durch den Einsatz von IKT und Robotern im Unterricht erwerben kann, in zwei Teilen zusammen – zum einen nennt sie die Inhalte bzw. Fachdisziplinen, in denen fachliche Kompetenzen erworben werden, und zum anderen nennt sie akademische Kompetenzen, die übergeordnete Kompetenzen umfassen:

1. Fachdisziplinen
 - a. Physik
 - b. Biologie
 - c. Geographie
 - d. Elektronik
 - e. Mathematik
 - f. Mechanik
 - g. Naturwissenschaft allgemein
2. Akademische Kompetenzen
 - a. Schreiben
 - b. Lesen
 - c. Recherche
 - d. Kreatives Denken
 - e. Gruppenarbeit, kooperatives Arbeiten
 - f. Entscheidungen treffen
 - g. Probleme lösen
 - h. Kommunikation

(Eguchi, 2007, S. 2543)

Der UNESCO Bericht „Recognizing the potential of ICT in early childhood education“ zeigt Beispiele auf, wie diese Kompetenzen gefördert werden können. (Kalaš, 2010, S. 28 ff.)

Generell ist der Literatur zu entnehmen, dass sich Kinder und Jugendliche ihren Lernprozess durch den Einsatz von Robotern selbstbestimmt, durch Beobachtung und eigenes Handeln gestalten können, so auch bei Silva et al. (2008). Das Lernen geschieht vor allem durch den kollaborativen Prozess, wenn Lernende und Lehrende gemeinsam an einer Aufgabenstellung arbeiten. Roboter im Unterricht motivieren die Lernenden zum Denken, zum Strukturieren ihrer Ideen, zum Aufstellen von Hypothesen und zum Erreichen von Zielen. (Silva et al., 2008, S. 132)

Der Einsatz von Bodenrobotern – Roboter, die direkt am Gerät selbst programmiert werden können und dafür keine weitere Software oder Computer benötigen und vor allem für den Einsatz am Boden oder Tisch konstruiert wurden – hilft Kindern beim Ausprägen von Kompetenzen in den Bereichen:

- Kennen der Zahlen,
- Zählen und logisches Denken,
- Lösen topologischer Probleme im Raum,
- Problemlösungsorientiertes Lernen, aber auch Inquiry-Based Learning.

(Beraza, Pina & Demo, 2010, S. 553), (Mittermeir, 2010, S. 59 ff.)

3.2.1 Interaktion mit Objekten und Lernen

Die Begriffsentwicklung und die Fähigkeit Dinge bei ihrem Namen zu nennen etabliert sich beim Kind in den ersten Lebensjahren (Gelman & Kalish, 2006, S. 724), wobei hier vor allem

in den ersten beiden Lebensjahren wesentliche Grundsteine für die kognitive Leistungsfähigkeit der jungen Menschen gelegt werden. (Brito & Barr, 2012, S. 812). Brito & Barr (2012) sehen Kinder, die zweisprachig aufwachsen im Vorteil, da diese über die kognitive Flexibilität verfügen, sich leichter in neuen Themen zurecht zu finden. (Brito & Barr, 2012, S. 812)

Burnett Heyes, Zokai, van der Staaij, Bays & Husain (2012) zeigen nicht nur auf, dass sich das visuelle Kurzzeitgedächtnis mit zunehmendem Alter weiterentwickelt und einhergehend die Fähigkeit steigt, sich eine Anzahl unterschiedlicher Elemente zu merken. (Burnett Heyes et al., 2012, S. 528) Sie arbeiten auch heraus, dass sich mit zunehmenden Alter die Präzision beim Merken von unterschiedlichen Elementen erhöht (Burnett Heyes et al., 2012, S. 536), ohne dass sich die Anzahl der diskreten Objekte vermindert. (Burnett Heyes et al., 2012, S. 537) Durch Hinzeigen auf Objekte können Kinder auf diese aufmerksam machen und legen damit die Basis für die menschliche Kommunikation. (Matthews, Lieven & Tomasello, 2007, S. 1744)

Die Grundlagen zum Verständnis von Vorgängen und Objekten im realen Leben entwickeln Kinder durch Erleben und Angreifen selbiger. Die physikalische Auseinandersetzung mit der direkten Umwelt eines Kindes hilft dem Kind dabei, Verständnis und Wissen aufzubauen, sowie auch die Lese- und Schreibkenntnisse zu fördern. (Jensen, Burleson & Sadauskas, 2012, S. 50) Hierbei ist interessant zu beobachten, dass nach Bruner die Entwicklung des Kindes in drei chronologische Stufen zu unterteilen ist: Als erstes sind es Handlungen, die Kinder nutzen um Wissen zu erfahren bzw. zu repräsentieren. In der nächsten Entwicklungsstufe sind es Bilder oder Piktogramme, mit denen Wissen repräsentiert werden kann. Zum Abschluss sind es Symbole, Zahlen oder Buchstaben, die zum Wissenserwerb oder zur Wissensrepräsentation eingesetzt werden. (Jensen, Burleson & Sadauskas, 2012, S. 51 f.) Die Entwicklung der Bedienung eines Computers verlief parallel in genau entgegengesetzter Richtung. Wo früher noch reine Textkonsolen mit Symbolen, Zahlen und Buchstaben zur Repräsentation von Informationen zum Einsatz kamen, sind heute überwiegend grafische Benutzeroberflächen mit Bildern und Piktogrammen zu finden. In letzter Zeit werden zunehmend neue Eingabemöglichkeiten für Computersysteme entwickelt, die durch physikalische Interaktionen, wie Berührungen oder Manipulation von physikalischen Objekten eine Bedienung ermöglichen. Es werden sog. Tangible Interfaces oder Tangible User-Interfaces (TUI) entwickelt. (Jensen, Burleson & Sadauskas, 2012, S. 52). Mehr zu Tangible Interfaces in Kapitel 4.3.2.

Der Einsatz von Tangible Interfaces erlaubt eine effektive Förderung von Kindern. Tangible Interfaces unterstützen die Ausprägung von Lese- und Schreibfähigkeiten durch multimodales Lernen, was vor allem auch in Gruppen zu sehr guten Lernerfolgen führen kann und somit ein Agieren in Vygotskys „Zone of proximal development“ (siehe Kapitel 2.1.5) ermöglicht. (Jensen, Burleson & Sadauskas, 2012, S. 53)

Um allfälligen Schwächen in diesen Bereichen vorzubeugen, empfiehlt es sich, erste Aktivitäten zur Ausprägung von Lese- und Schreibkompetenzen schon vor der Schulpflicht, eher schon zu Beginn des Kindergartenalters zu setzen. (Lonigan, Allan & Lerner, 2011, S. 499) Der Einsatz von Computern unterstützt Kinder beim Schreibenlernen, insbesondere beim Erlernen der Zeichensetzung und des Abgrenzens von Wörtern durch das Setzen von notwendigen Leerzeichen. (Yost, 2003, S. 111) Nach Armstrong (2011) korreliert eine ausgeprägte Visuomotorik stark mit den Leistungen beim Lesen und in Mathematik. Unter Visuomotorik versteht man die Fähigkeit eine Form oder ein Objekt wahrzunehmen und diese Form bzw. dieses Objekt wiederum auf Papier zu replizieren. (Armstrong, 2011, S. 3677) Die Kombination von Technologie und individueller Arbeit mit dem Kind ermöglicht eine Förderung und Steigerung der Visuomotorik. (Armstrong, 2011, S. 3680)

3.2.2 Problemlösen

Kazakoff, Sullivan & Bers (2013) verwenden die Definition des Computerprogrammierens von Liao und Bright, wonach dieses definiert ist als "using [...] sequence of instructions, variables, recursion, etc. to write solutions to problems [...]". (Liao & Bright, 1991, S. 253, zitiert in: Kazakoff, Sullivan & Bers, 2013, S. 247)

Da der Sequenzierung in Hinblick auf die Fähigkeit Computerprogramme zu erstellen und zu verstehen hohe Bedeutung beigemessen wird, haben Kazakoff, Sullivan & Bers (2013) in einer Studie, bei der sie Roboter im Kindergarten eingesetzt haben, die Auswirkungen auf die Fähigkeiten zur Sequenzierung bei jungen Kindern untersucht. Sequenzierung, als Teil der Planung, beinhaltet das Anordnen von Objekten und Aktionen in der richtigen Reihenfolge. Als Beispiele nennen die Autorinnen das Nacherzählen einer Geschichte im richtigen Ablauf, das Anordnen von Zahlen in der richtigen Reihenfolge und das Nacherzählen der Aktivitäten eines Tages. (Kazakoff, Sullivan & Bers, 2013, S. 247) Der Einsatz von Robotik und Programmieraktivitäten scheint nach den Ergebnissen von Kazakoff, Sullivan & Bers (2013) die Fähigkeit zur Sequenzierung bei jungen Kindern zu fördern. (Kazakoff, Sullivan & Bers, 2013, S. 251 f.) Weiters fördert der Einsatz von Programmiersprachen für Kinder das logische Denken und die Problemlösungskompetenzen der Kinder. (Smith, Cypher & Tesler, 2000, S. 78 & 80 f.)

„Bildung in jeder Form soll zum Problemlösen befähigen.“ (Schubert & Schwill, 2011, S. 81) Schubert und Schwill (2011) sehen das Fach Informatik als prädestiniert für die Ausbildung von Problemlösungskompetenzen an. In den traditionellen Unterrichtsfächern fehlt es an Möglichkeiten sich den Grundlagen der Problemlösung zu widmen. Im Unterrichtsfach Informatik entsteht für die Schülerinnen und Schüler die Gelegenheit ihre eigenen Problemlösungsstrategien zu entwickeln und anzuwenden. Dabei sind die Problemlösungen so konkret zu beschreiben, dass damit nicht nur eine Lösung geliefert wird und Maschinen diese auch ausführen können, sondern dass diese auch anderen zugänglich gemacht werden können. (Schubert & Schwill, 2011, S. 81)

3.2.3 Räumliche Orientierung

Radu & MacIntyre (2009) berichten von drei Referenzsystemen, die Erwachsenen bei der Verarbeitung räumlicher Beziehungen zur Verfügung stehen: der egozentrierte Modus, der objektzentrierte Modus und der umweltzentrierte Modus. Beim egozentrierten Modus dient der eigene Körper bei Richtungen und Abständen als Referenzpunkt (etwas liegt links von mir), beim objektzentrierten Modus nimmt ein externes Objekt und nicht der eigene Körper die Rolle des Referenzpunkts ein (etwas liegt rechts neben dem Objekt), wohingegen beim umweltzentrierten Ansatz die Umgebung als Referenzpunkt herangezogen wird (etwas befindet sich im Norden). (Radu & MacIntyre, 2009, S. 211)

Die Orientierung im Raum spielt in der menschlichen Evolution eine wichtige Rolle. Da die räumliche Orientierung essentiell für das Überleben ist, ist es für die Wissenschaft sehr interessant herauszukristallisieren, welche Fähigkeiten zu Beginn des Lebens zur Verfügung stehen und daher eine zentrale Rolle in der Herkunft des Wissens der Menschen spielen. (Newcombe & Huttenlocher, 2006, S. 734) Bislang bestehen zwei wesentliche Gruppen von Modellen zum Verständnis der Funktionsweise der räumlichen Orientierung: betrachtet man die modularen Modelle, so werden die unterschiedlichen Quellen räumlicher Informationen unabhängig von einander in separaten kognitiven Prozesseinheiten verarbeitet. Die alternativen Modelle sehen eine Bündelung der räumlichen Informationen vor, wobei bei der Kombination selbiger die unterschiedlichen Quellen basierend auf ihrer Nützlichkeit gewichtet werden. (Newcombe & Huttenlocher, 2006, S. 735) Darüber hinaus zeigen Menschen unterschiedliche Ausprägungen in den Fähigkeiten der räumlichen Wahrnehmung und Orientierung. (Newcombe & Huttenlocher, 2006, S. 735 f.)

Kinder entwickeln bereits in den ersten Jahren die Fähigkeit sich räumlich zu orientieren. Eine Grundform der objektzentrierten Sichtweise entwickelt sich als Erstes: Kinder können Objekte in ihre Bestandteile (wie vorne und hinten) zerlegen und angeben, ob sich ein

anderes Objekt in der Nähe dieser Teile befindet. Der egozentrierte Modus zur räumlichen Orientierung entwickelt sich danach, ungefähr mit 5 Jahren und eine vollständige objektzentrierte räumliche Orientierung mit etwa 12 Jahren, wo die „Sichtlinie“ von Objekten beschrieben werden kann. (Radu & MacIntyre, 2009, S. 211)

3.2.4 Soziale Kompetenz

Peers, gleichaltrige Personen, bieten ein gutes Umfeld für die Ausprägung von sozialen Kompetenzen. Es entwickeln sich hier Kompetenzen zur Kameradschaft, Unterhaltung und zum zwischenmenschlichen Lernen. Die Interaktion von Kindern untereinander fördert die Entwicklung kritischer sozialer Fähigkeiten, das zwischenmenschliche Verständnis und hebt das soziale Selbstwertgefühl. Diese Kompetenzen sind frühzeitig zu erlangen und zu fördern, denn in der Schule stehen Kinder nach Bierman & Erath (2006) sonst ernsthaften Problemen mit ihren Mitschülerinnen und Mitschülern gegenüber. (Bierman & Erath, 2006, S. 595) Früh geförderte soziale Kompetenzen unterstützen Kinder dabei Freundschaften zu entwickeln und sich von sozialen Rückschlägen schneller zu erholen. (Bierman & Erath, 2006, S. 596)

Kinder mit ausgeprägten sozialen und emotionalen Kompetenzen verfügen zusätzlich über bessere Beziehungen zu ihren Eltern und ihren Lehrerinnen und Lehrern. Eine frühe Intervention bei sozialen und emotionalen Problemen hilft dabei, spätere Folgen, die auf mangelnd ausgeprägten sozialen und emotionalen Kompetenzen beruhen, abzuwehren. (McCabe & Altamura, 2011, S. 513) Einige Mängel im Bereich der sozialen und emotionalen Kompetenzen lassen sich aus der Qualität der Eltern-Kind-Beziehung ableiten. Eine unsichere Eltern-Kind-Beziehung ist geprägt von Misstrauen, mangelnder Fürsorge und fehlender Unterstützung. (McCabe & Altamura, 2011, S. 515) Kinder, die in frühen Jahren eine hochqualitative Fürsorge erfahren, werden wahrscheinlich ein besseres geistiges und sozioemotionales Wachstum erleben. (McCabe & Altamura, 2011, S. 535)

Virtuelle Welten können junge Menschen auf ihrem Weg zum Erwachsenen-Dasein unterstützen. Sie bieten Gelegenheiten in (virtuellen) Gemeinschaften soziale, emotionale und kognitive Fähigkeiten zu entwickeln, wie es auch grundsätzlich in Klassen und auf Spielplätzen geschieht. Darüber hinaus wird dem Aspekt „user-generated content“, also von Userinnen und Usern erstellte digitale Inhalte, zunehmend Rechnung getragen, was in weiterer Folge die IT- und Medienkompetenzen der beteiligten Jugendlichen fördert. (Beals, 2010, S. 46)

3.2.5 Kompetenzentwicklung im Kindergartenalter

Morgado, Bulas Cruz & Kahn (2003) vertreten die Ansicht, dass Computer bereits im Kindergartenunterricht (bei 3- bis 5-jährigen Kindern) eingeführt werden sollen, genauso wie andere Bereiche des modernen Lebens Einzug in den Kindergartenalltag gefunden haben. Kinder sollen mit Computern im Allgemeinen vertraut sein, und nicht nur mit einer Auswahl an speziellen Fertigkeiten in der Bedienung. Sie sehen Programmierung als einen Weg an, Kindern die Möglichkeit zu geben, grundlegende Konzepte des Computers zu verstehen und der Furcht vor der Bedienung von Technologie oder der Entwicklung einer ablehnenden Haltung gegenüber dieser entgegenzuwirken. Kinder sollen erkennen, dass Computer anpassbare Werkzeuge sind, die uneingeschränkt im Alltag zum Einsatz kommen können. (Morgado, Bulas Cruz & Kahn, 2003, S. 178)

Auf den ersten Blick erscheint bei einer Einführung in die Computerprogrammierung für Kindergartenkinder die Hürde, dass sie noch über keine oder nur schwach ausgeprägte Lese- und Schreibfähigkeiten verfügen. Es gibt einige Ansätze um diese Herausforderung zu meistern. Jedoch, so Morgado, Bulas Cruz & Kahn (2003), warum soll man sich damit beschäftigen? Welchen Zweck kann Programmieren im Kindergartenalter erfüllen? Der Kindergarten zielt darauf ab, die Grundfähigkeiten der Kinder zu erweitern und stärker ausprägen. Dazu zählen: die persönliche und soziale Entwicklung ebenso wie Ausdruck und Kommunikation (motorischer, künstlerischer und musikalischer Ausdruck, gesprochene

und geschriebene Sprache, Mathematik) und Wissen über die Welt. Kinder müssen ermächtigt werden, so leicht und umfassend wie möglich von ihrer Umwelt, sowie ihren eigenen Erlebnissen zu lernen. (Morgado, Bulas Cruz & Kahn, 2003, S. 178)

Computer gehören in der heutigen Gesellschaft zum Alltag; aber es geht im Kindergarten nicht nur um das Erlernen von Grundfertigkeiten im Zusammenhang mit bestimmter Software, sondern darum, die ganze Leistung eines Computersystems zu erschließen. Generell sollen Kinder ein Level „technologischer Geläufigkeit“ erlangen, ähnlich der Lesegeläufigkeit, die heutzutage als Grundfähigkeit angesehen wird. (Morgado, Bulas Cruz & Kahn, 2003, S. 178).

„[...] we believe that computational thinking supported through educational programming is a valid contribution to the general primary and secondary education for all – not because we want to attract young people into university Computer Science programmes, but because it constitutes important part of so-called skills for the 21st century learning.“ (Kalaš, 2014, S. 29)

Morgado, Bulas Cruz & Kahn (2003) sehen in folgenden Bereichen in Hinblick auf Informatik und Programmierung in Kindergärten den größten Forschungsbedarf: Es gilt geeignete Konzepte und Aufgabenstellungen zu entwickeln, so dass Kinder im Alter von 3 bis 5 Jahren selbständig beginnen können, die Welt der Programmierung und Informatik zu erkunden. Mit zunehmendem Schwierigkeitsgrad werden Hilfestellungen von Lehrkräften notwendig. Hier gilt es zu klären, wie die dafür notwendige Zeit im Kindergarten aufgebracht werden kann, um die Ausprägung von IKT-Kompetenzen zu ermöglichen und zu fördern. (Morgado, Bulas Cruz & Kahn 2003, S. 185)

3.3 Higher-Order Thinking und Computational Thinking

In der Literatur werden häufig die Begriffe „Higher-Order Thinking“ und „Higher-Order Rules - Problem Solving“ genannt. Bei „Higher-Order Rules“, Regeln höherer Ordnung, handelt es sich um die Repräsentation von Kombinationen einfacher Regeln, die zum Lösen von komplexen Problemen eingesetzt werden können. „Higher-Order Thinking“, Denken höherer Ordnung, umfasst mehr, als nur einfache Regeln oder das Abrufen von Informationen. Es ist eine Interaktionsfunktion zwischen kognitiven Strategien, Metakognition und nicht strategischem (domänenspezifischen) Wissen zur Lösung neuer Probleme. Dieses Denken höherer Ordnung ist zielgerichtet, mehrstufig und verfolgt ein prozessähnliches Vorgehen, mit Gestaltung, Entscheidungsfindung, Problemlösung und Evaluation. (Lim & Tay, 2003, S. 427)

„Computational Thinking“ beschreibt eine Bandbreite von analytischen und problemlösenden Fähigkeiten, Haltungen, Gewohnheiten und Vorgehen, die in der Informatik benötigt werden, sein Begründer ist Seymour Papert (siehe auch Kapitel 2.1.4). Computational Thinking umfasst auch die Fähigkeit eine Programmiersprache vom Verhalten eines Computers zu abstrahieren und mögliche Fehler zu identifizieren. Ein Curriculum, welches Robotik- und Programmieraktivitäten beinhaltet, kann junge Kinder im Computational Thinking fördern. (Sullivan & Bers, 2015, S. 3), (Bers, Flannery, Kazakoff & Sullivan, 2014, S. 156)

Wing (2008) beschreibt den Begriff „Computational Thinking“ umfassender. „Computational Thinking“ hat mit dem mathematischen Denken die grundsätzliche Herangehensweise zur Problemlösung gemein. Mit dem ingenieurwissenschaftlichen Denken teilt es sich wie große, komplexe Systeme, welche in den Rahmenbedingungen der realen Welt operieren, gestaltet und bewertet werden können. Mit dem wissenschaftlichen Denken hat „Computational Thinking“ die Ansätze gemeinsam, wie wir Verständnis über Berechenbarkeit, Intelligenz, Verstand und das menschliche Verhalten erlangen können. Die Grundelemente des „Computational Thinking“ sind die Abstraktion und Automatisierung. (Wing, 2008, S. 3717) Nach Wing (2008) wird Computing von mehreren Dimensionen, die zueinander in Wechselwirkung stehen, vorangetrieben. Das sind zum einen die Gesellschaft selbst, zum anderen die Technologie(entwicklung) und die Wissenschaft. (Wing, 2008, S. 3722)

Diese Darstellung des Begriffs inkludiert nicht nur die Repräsentation einer Lösung für ein Problem, so dass diese ein Computer verstehen kann, sondern auch andere Menschen. (Manches & Plowman, 2015, S. 6)

„The inclusion of humans as ‚information processing agents‘ broadens the notion of computational thinking to allow for a graduation from general communication skills to the specific practice of coding and allows for familiar activities such as giving the robot instructions to make a jam sandwich.“ (Manches & Plowman, 2015, S. 6)

Die Vielzahl unterschiedlichen Publikationen zu dem Thema und der auch zum Teil diversen Auseinandersetzung damit, lässt oftmals den Schluss aufkommen, dass es sich bei „Computational Thinking“ um das Programmieren eines Computersystems oder Roboters handelt. Das es jedoch deutlich mehr ist, zeigt die Tatsache auf, dass zum Erlernen des „Computational Thinking“ nicht zwangsweise ein Computer von Nöten ist. (Voogt, Fisser, Good, Mishra & Yadav, 2015, S. 716) Programmierung ist jedoch nur ein Aspekt der mit Computer und Informatik sowie „Computational Thinking“ in Verbindung gebracht werden kann. Informatik ist die Disziplin und der Bereich aus dem das „Computational Thinking“ entstanden ist, gleichwohl die Informatik nicht die einzige Disziplin ist, die diese übergeordneten Denkstrukturen und -muster benötigt. Der Fokus des „Computational Thinking“ liegt auf den zuvor erwähnten höheren Konzepten, die erlernt und in unterschiedlichen Domänen angewandt werden können. (Voogt et al., 2015, S. 718)

Die ACM stellt auf ihrer Webseite¹⁰ der „Computer Science Teachers Association“ zahlreiche weiterführende Informationen und Lernmaterialien zu „Computational Thinking“ zur Verfügung. (Seiter & Foreman, 2013, S. 61) Im Rahmen der Kompetenzorientierung formulierten Atmatzidou & Demetriadis (2016) in ihrer Studie Kompetenzen, die ausgewählten Grundkomponenten des „Computational Thinking“ wie Abstraktion, Generalisierung, Algorithmen, Modularität und Dekompositionen zugeordnet werden können. (Atmatzidou & Demetriadis, 2016, S. 664)

Eine mögliche Forschungslücke in diesem Themengebiet und im Kontext der gegenständlichen Arbeit wird in Kapitel 6.2 behandelt.

¹⁰ <http://csta.acm.org/Curriculum/sub/CompThinking.html>, 23.04.2018

4 3- bis 6-jährige Kinder lernen Programmieren

„Das Gedankengut des Programmierens darf nicht zum Geheimkult der Informatikspezialisten [und -spezialistinnen, Anm. d. Verf.] verkümmern.“ (Nievergelt, 1999, S. 364)

Diesem Appell von Nievergelt (1999) folgend behandelt dieses Kapitel zu Beginn den Einstieg in die Programmierung (Kapitel 4.1) und mögliche Ansätze und Herausforderungen, die es beim Programmieren lernen zu berücksichtigen und zu meistern gilt. (Kapitel 4.2) Grundsätzlich gilt es im Rahmen der gegenständlichen Arbeit zu beachten, dass Kinder im Alter von 3 bis 6 Jahren nicht oder kaum lesen können. Es sind daher Programmierumgebungen und Werkzeuge zu wählen, die mit wenig bis gar keiner Schrift auskommen, damit Kinder im Alter von 3 bis 6 Jahren diese selbständig bedienen und nutzen können.

Das Konzept des Programmierens ist schwierig zu beschreiben, es unterliegt ständigen Veränderungen und hat viele Bedeutungen für verschiedene Personen unterschiedlicher Professionen. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 132) Als elementare Programmierung beschreiben Kabátová, Kalaš & Tomcsányiová (2016) Aktivitäten, bei denen junge Kinder durch die Steuerung von Agenten Problemstellungen lösen oder deren zukünftiges Verhalten planen können – das alles in einer digitalen Umgebung. Die Autorinnen und Autoren sind der Überzeugung, dass elementare Programmierung ein guter Weg ist, um Problemlösungsfähigkeiten im Bereich des Computational Thinkings auszuprägen. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 131) Das Themengebiet der elementaren Programmierung kann nach Kalaš (2016) in drei Dimensionen betrachtet werden. Dazu gehören Akteure, Art der Steuerung und Interaktionsgrad. (Kalaš, 2016, S. 4)

„Programming, at its core, is about giving instructions – or commands – to be executed by a machine. Clearly, the machine needs not to be a computer. It can be a robotic device or a set of ‘smart bricks’. And the commands need not be typed on a keyboard, but can take the form of components to be assembled manually, icons to be snapped into place, and increasingly, voice, gesture, force-feedback.“ (Ackermann, 2012, S. 5)

Der Einstieg in den Überblick über Angebote an Hard- und Software am Markt für Programmieranfängerinnen und -anfänger erfolgt durch Vorstellung von Lernumgebungen zum Programmieren für die Zielgruppe. Da junge Kinder gerne Dinge „erleben“ und vieles in die Hand nehmen möchten, eignen sich gerade für diese Zielgruppe sog. Tangible Interfaces sehr gut. Neben einer Vorstellung von Produkten und Projekten in diesem Bereich, stellt die Autorin Tangible Interfaces grafischen Benutzeroberflächen gegenüber. (Kapitel 4.3) Ein weiteres wichtiges Thema im Kontext der frühen Technikbildung ist Educational Robotics. (Kapitel 4.4)

4.1 Einstieg in die Programmierung

„[Seymour Papert advised:, Anm. der Verf.] not to teach children to program for the sake of programming. Instead, he said: Use the knowledge of programming to create contexts where other play and learning can happen. Children will engage in programming if they get something out of it right now – not later when they’ll grow up!“ (Ackermann, 2012, S. 9)

Ackermann (2012) stellt vor, wie Programmieren im Kontext von jungen Kindern als Mittel für: „1) making things do things (instruct a device to follow and execute orders); for 2) ‘animating’ things (endow a device with a ‘mind of its own’, teach it to ‘look out for itself’); and for 3) ‘poking’ things (modulate how things act or interact by tweaking some parameters in their environment).“ angesehen und verwendet werden kann. (Ackermann, 2012, S. 5)

Kabátová, Kalaš & Tomcsányiová (2016) nennen Basisinhalte, die während des Erlernens der Programmierung vermittelt werden sollen:

- Problemlösen und mit Lösungen arbeiten
- Steuerung eines Agenten

- Direkte Steuerung eines Agenten
- Indirekte Steuerung eines Agenten (Aufbau und Handhabung zukünftiger Verhaltensweisen)
- Fortführende Konzepte der Grundlagen der Programmierung (z.B. Parameter, Schleifen, Prozeduren, etc.)
- Basteln mit interaktiven Umgebungen
 - Viele Agenten und deren Eigenschaften
 - Statische Szenarien
 - Dynamische Szenarien

(Kabátová, Kalaš & Tomcsányiová, 2016, S. 133)

Ergänzend führt Ackermann (2012) drei Gründe und deren Implikationen für das Programmierenlernen junger Kinder an:

1. Dinge beherrschen – übernehmen, überlassen, übernehmen
Kinder sind bei der Programmierung in der Lage, über eine Welt die Kontrolle zu übernehmen. Dazu erstellen und kontrollieren sie Dinge, die ihre Aufträge ausführen. Da die Dinge, die von den Kindern erteilten Aufträge zuverlässig ausführen, lernen die Kinder loszulassen und zu delegieren.
2. Dinge animieren – erstellen, animieren, interagieren
Durch das Bauen und Spielen mit Dingen, die sich so verhalten, als hätten sie einen eigenen Willen, lernen Kinder auf welche Weise animierte Dinge ihr Verhalten regulieren und miteinander interagieren. Auf spielerische Art und Weise lernen die Kinder zwischen selbst- und fremdgesteuerter Kontrolle zu unterscheiden.
3. Dinge modulieren – so nehmen wie es ist, modifizieren, belassen
Kinder verwenden Dinge, die sie vorfinden, modifizieren oder mischen sie mit anderen und verleihen den Dingen so neue Fähigkeiten, um sie in einem neuen Kontext einsetzen zu können. Dies geschieht iterativ und erlaubt den Kindern kontinuierlich das Konzept der Wiederverwendung zu erlernen.

(Ackermann, 2012, S. 8)

Um in die Welt der Programmierung einsteigen zu können, gibt es zwei grundlegende Konzepte, die in der Wissensvermittlung zu Beginn angewandt werden. Zum einen ist dies das Denken in Variablen, die zu einem definierten Zeitpunkt einen bestimmten Wert annehmen. Als Metapher dient hier ein Container, der zu einem Zeitpunkt x mit Inhalt y gefüllt ist. Zum anderen sind das Zustände bzw. Zustandsautomaten, die in ihren unterschiedlichen Komplexitätsgraden ideal sind, um die Zustände eines Programmes zu repräsentieren. Die Herausforderung ist jedoch, den Lernenden den Unterschied näher zu bringen zwischen etwas zu sein bzw. etwas zu haben und sich in einem bestimmten Zustand zu befinden. (Weigend, 2008, S. 151 f.) Die Denkweise in Automaten bzw. Zustandsautomaten bringt zahlreiche Vorteile für das Programmierenlernen mit sich. Die Lernenden sind dadurch in der Lage, Konstrukte von Programmiersprachen zu nutzen um sinnvolle Zustände durch Variable und Objektattribute mit zugewiesenen Daten abzubilden, Methoden aufzurufen und mehrere Automaten in ein (komplexes) Gesamtsystem zusammenzuführen. Die so erworbenen Kompetenzen helfen die logische Korrektheit von komplexen und nicht mehr intuitiven Programmen zu prüfen und zu verstehen. (Weigend, 2008, S. 160)

Eine grundsätzliche Variante, wie Algorithmen Studierenden und gleichermaßen auch jungen Kindern nähergebracht werden können, zeigt Futschek (2007) auf: Lernende repräsentieren die Elemente eines Algorithmus und „spielen“ gemeinsam in der Realität einen Algorithmus durch. Dazu sind weder Programmierkenntnisse noch Computer von Nöten. (Futschek, 2007, S. 2 ff.) Es wird das algorithmische Denken trainiert und durch die gemeinsame

Aktivität und die zwischenmenschlichen Interaktionen entstehen neue Ansätze und Denkmuster, die helfen, informatische Probleme zu lösen und weiterführende Konzepte wie Parallelisierung, Synchronisierung oder das Teile & Herrsche-Prinzip (Divide & Conquer) kennen zu lernen und zu erleben. (Futschek, 2007, S. 6)

Kolczyk (2008) fordert, dass neue didaktische Methoden in die Ausbildung der Informatiklehrkräfte einfließen müssen, um diese tatsächlich auch in der Praxis einsetzen zu können. (Kolczyk, 2008, S. 270) Das Erleben und „Spielen“ von Algorithmen im Unterricht muss wesentlich stärker gelebt werden, als das bloße Vorzeigen und Ausführen. (Kolczyk, 2008, S. 267) Durch einen Wechsel der Perspektive, weg von der operationalen Seite hin zu einer deskriptiven Sicht, kann dies Lehrenden gut gelingen. (Futschek, 2005, S. 327 f.)

Futschek & Moschitz (2011) beschreiben die Entwicklung eines realen Spieles, welches das algorithmische Denken fördern soll. Für Kinder ab 5 Jahren soll „Tim the Train“ als Basis zur Entwicklung der algorithmischen Denkfähigkeiten dienen. Die Idee zu „Tim the Train“ basiert auf dem „Bin Packing Problem“, einem bekannten Problem der Informatik – wie kann ein Rucksack optimal gepackt werden, ohne dass er dabei überladen und dennoch der zur Verfügung stehende Platz sehr gut genutzt wird. Bei „Tim the Train“ sind Waggons eines Zuges (aus Holz) mit Bausteinen zu füllen und zwar so, dass der zur Verfügung stehende Platz nicht überladen wird und der Waggon dennoch möglichst voll wird. Ebenso müssen alle zur Verfügung stehenden Steine in die Waggons geladen werden. (Futschek & Moschitz, 2011, S. 156 f.)

4.2 Ansätze und Herausforderungen

Die aktive Auseinandersetzung mit Programmierung verhilft Kindern zu einem Rollenwechsel, weg von der Konsumentin und dem Konsumenten von Computern, hin zu Designerinnen und Designern, die die Möglichkeiten, die ein Computer bietet, ausschöpfen können. Programmieren lernen hilft, Problemlösungen zu erarbeiten und Lösungsstrategien zu entwickeln. (Resnick, Flanagan, Kelleher, MacLaurin, Ohshima, Perlin & Torres, 2009, S. 3294 f.), siehe u. a. (Wang & Chen, 2010, S. 219)

Programmieren lernen, eventuell auch mit Robotern und anderen Materialien, wird in vielen Ländern seit den 1990er Jahren schon als notwendig und alltäglich betrachtet, so auch u.a. in Norwegen, Großbritannien und den Vereinigten Staaten von Amerika. (Lau, Tan, Erwin & Petrovic, 1999)

Die meisten verfügbaren Angebote sind jedoch für Kinder ab 6 Jahren oder älter. So wie z.B. spezielle Sommer-Camps. Steinová & Sisková (2010) berichten über ein erfolgreiches einwöchiges Konzept von Informatik-Camps, wo neben der Vermittlung theoretischer Grundlagen der Informatik auch zahlreiche Programmier-Workshops, eingebettet in ein sportliches und soziales Rahmenprogramm, das den Zusammenhalt der Gruppe fördern soll, stattfinden. (Steinová & Sisková, 2010, S.171 f.)

Eine Herausforderung, auch noch für ältere Kinder, ist die explizite und direkte Benennung von Elementen im Zusammenhang mit Algorithmen und Programmieren, gerade wenn diese Elemente abstrakt oder zu komplexeren Teilen zusammengefasst sind. (Weigend, 2010, S. 192) Auch Kabátová, Kalaš & Tomcsányiová (2016) sind sich bewusst, dass junge Kinder über kein ausgeprägtes abstraktes Denken verfügen und dieses schrittweise entwickelt und gefördert werden muss. Dennoch gibt es viele Aktivitäten, die Programmierenlernen unterstützen, ohne dass Abstraktionsfähigkeiten erforderlich sind. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 131)

„We believe that learning to think computationally and to program one’s solutions can be done gradually by doing specific activities that only slowly lead to a true abstraction, decomposition of problems and generalization of solutions.“ (Kabátová, Kalaš & Tomcsányiová, 2016, S. 131)

Um die Benennung und Repräsentation der Elemente zu erleichtern, schlagen Fernaeus, Kindborg & Scholz (2006) vor, Programmierumgebungen für Kinder um Comic-ähnliche Symbole zu bereichern. Dabei sollen diese Comic-Symbole auch einen Bezug zum jeweiligen Kontext aufweisen. Der unmittelbare visuelle Kontext ermöglicht eine visuelle Direktheit bzw. Eindeutigkeit. Darüber hinaus zeigen die Comic-ähnlichen Symbole die wahrnehmbaren Aktionen und Eigenschaften der Objekte für Kinder besser auf und erleichtern somit das Verständnis des Kontexts und die Interpretation. (Fernaeus, Kindborg & Scholz, 2006, S. 128)

Nach Reichert, Nievergelt & Hartmann (2005) sind die grundlegenden Anforderungen an eine Programmiersprache für Anfängerinnen und Anfänger folgende: die Komplexität muss reduziert und versteckt werden; die Programmierung muss visualisiert werden und die einfache Programmierumgebung soll einen überschaubaren Sprachumfang aufweisen; die Aufgabenstellungen müssen alltagsorientiert sein. (Reichert, Nievergelt & Hartmann, 2005, S. 10 f.)

Am Beispiel von „Bebras International Contest on Informatics and Computer Literacy“ erläutern Dagiene & Futschek (2008) grundlegende Kriterien zur zielgruppengerechten Gestaltung von Aufgaben im Kontext der Informatik:

Gute Aufgaben ...

- stehen in Bezug zur Informatik, Informationswissenschaft oder der Bedienung von Computern.
- ermöglichen Lernerfahrungen.
- sind in drei Minuten lösbar.
- verfügen über unterschiedliche Schwierigkeitsgrade.
- sind zielgruppengerecht formuliert und gestaltet.
- sind unabhängig von nationalen Curricula.
- sind unabhängig von spezifischen IT-Systemen.
- verfügen über leicht verständliche Problembeschreibungen.
- sind auf einer (Bildschirm-)Seite darstellbar.
- sind auf einem Computer ohne zusätzliche Hardware, Papier und Stift lösbar.
- sind politisch korrekt.

(Dagiene & Futschek, 2008, S. 22 f.)

Darüber hinaus sollten Aufgaben lustig sein, über Bilder und / oder interaktive Elemente verfügen und Rückmeldungen zur Bearbeitung von Aufgaben sollten unmittelbar nach Abschluss der Aufgabe erfolgen. (Dagiene & Futschek, 2008, S. 23)

4.3 Lernumgebungen zum Programmieren für Kinder

Viele der im Laufe der Zeit entwickelten Roboter, Tangible Interfaces, Programmiersprachen und Angebote zum Programmieren lernen, entstammen Ideen und der Arbeit von Mitarbeiterinnen und Mitarbeitern des MIT Media Lab. Mit ihrer Arbeit versuchen sie die Lücke zwischen abstrakten Computern und der Lernfähigkeiten der Kinder zu schließen. (McNerney, 2004, S. 326) Blikstein (2013) liefert einen guten Überblick über die Entwicklungen von Tangible Interfaces sowie Robotik-Produkten für Kinder innerhalb der letzten fast 40 Jahren, beginnend in den 1980er Jahren, wo LEGO® Konstruktionen mit der Programmiersprache LOGO kombiniert wurden bis hin zu den 2000er Jahren, in denen, dank der raschen technologischen Entwicklung, interaktive Materialien und das „Makers-Movement“ den Markt beherrschen. (Blikstein, 2013, S. 175 ff.)

„In the process, students would learn about mathematical and scientific ideas and about the design process itself – an idea which, incidentally, came back to popularity with the “Maker”

movement. However, the ideas behind the Maker movement have incorporated into constructionist theories and implementations for at least 20 years.” (Blikstein, 2013, S. 174)

Die Maker-Bewegung spielt eine große Rolle im Prozess wirkungsstarke Ideen und ausdrucksfähige Medien an (Schul-)Kinder heranzubringen. Die Analogie zur Programmierung ist erkennbar: Die Technologie wird immer besser und leistungsstärker sowie zugänglicher. Die Fähigkeiten und Fertigkeiten selbige zu bedienen werden immer wertvoller und wichtiger. (Blikstein & Krannich, 2013, S. 613)

„What Logo did for programming – bringing complex mathematics to the reach of schoolchildren – fabrication laboratories can do for design and engineering.“ (Blikstein & Krannich, 2013, S. 613)

4.3.1 Software

Bereits seit den späten 1960er Jahren wird nach Programmiersprachen und -umgebungen geforscht, die es vorrangig Kindern ermöglicht, in die Welt der Programmierung einzutauchen, ohne sich auf komplexe und teilweise komplizierte Art und Weise mit Syntax und Sprachen beschäftigen zu müssen. (Kelleher & Pausch, 2005, S. 129 f)

4.3.1.1 Programmierungsumgebung vs. Mikrowelt

Bevor einzelne Programmiersprachen, Werkzeuge und auch Roboter näher betrachtet werden, sind zwei wesentliche Begriffe, Programmierungsumgebung und Mikrowelt, zu erläutern.

Eine Programmierungsumgebung ist ein Werkzeug, eine Software, die in der Lage sein muss, die Erstellung und Ausführung eines weiteren Programms zu ermöglichen. Das kann eine Kombination aus einem einfachen Texteditor zur Erstellung und einer Kommandozeile zur Ausführung von Programmen sein. Meistens handelt es sich dabei um eine IDE – Integrated Development Environment, die bei der Organisation der Programmkomponenten und Ressourcen unterstützt und weiterführende Funktionen wie Hervorhebungen von Syntax und Auto-Vervollständigung sowie z.B. visuelle Debugger, Test-Software, Dokumentationsgeneratoren und vieles andere mehr anbietet. (Pears, Seidman, Malimi, Manila, Adams, Bennedsen, Devlin & Paterson, 2007, S. 210)

Die physische Metapher in Mikrowelten soll dazu beitragen, den Unterschied zwischen den mentalen Modellen der Lernenden und der Programmiersprache zu verringern. Die dabei benötigte Komponente „Welt“ kann durch eine Visualisierung oder eine physische Umgebung repräsentiert werden. Mikrowelten unterscheiden sich zum einen in den Programmiersprachen (dies können eigens entwickelte oder vereinfachte Versionen bereits bestehender sein), die sie verwenden, in der Art und dem Umfang ihrer eingesetzten Metaphern (die Metaphern können vorgegeben sein oder selbst gestaltet werden) und zum anderen im Grad ihrer Unterstützung beim Programmieren (hier sind die größten Unterschiede zu beobachten, von Programmbibliotheken, die in bestehende Umgebungen eingebunden werden müssen, bis hin zu vollständigen eigens entwickelten IDEs). (Pears et al., 2007, S. 211)

Mikrowelten eignen sich auch, gerade im konstruktivistischen Lernen, für entdeckendes Lernen. In Kombination mit game-based-learning Ansätzen können Mikrowelten hierbei beim konstruktiven Problemlösen und sozialen Lernen unterstützen. (Judmaier & Pohl, 2012, S. 110), siehe auch (Pohl, Rester & Judmaier, 2009) Kabátová, Kalaš & Tomcsányiová (2016) liefern eine Übersicht über Mikrowelten für den Einsatz in der Volksschule, in der diese nach Features und Darstellungen sowie Manipulationsmöglichkeiten klassifiziert sind. In der Übersicht und Vorstellung enthalten sind Thomas the Clown, Ice Cubes, Bee Tasks, World of Ant, EasyLogo, Baltie, Scratch und BeeWorld. (Kabátová, Kalaš & Tomcsányiová, 2016, S. 145)

Ein Hauptanliegen der Forschung im Bereich der Programmierung für Kinder ist es, Darstellungen zu entwickeln, die Kinder leicht verstehen und mit denen sie sich und ihre

Ideen ausdrücken können. (Fernaesus & Tholander, 2006, S. 1014) Kindern fällt programmieren leichter, wenn sie Programmierumgebungen nutzen, die sehr grafisch sind. Idealerweise erfolgt die Darstellung der Konstrukte der Programmiersprache in Form von Grafiken und durch eine ergänzende textbasierte Darstellung wird das Verständnis der Kinder für den Programmiercode gefördert. (Louca, 2004, S. 130) Die Mehrheit der Angebote am Markt für die Zielgruppe der 3- bis 6-jährigen Kinder ist wohl eher als Lernsoftware zu klassifizieren, denn als Programmierumgebung:

„Most computer programming tools target children at least 7-8 years old, but typically older. The plethora of mobile apps, software, and stand-alone technologies intended for younger children tend to focus on basic academic skills, such as recognition of letters and numbers, rather than content creation or high-level thinking.” (Flannery, Kazakoff, Bontá, Silverman, Bers & Resnick, 2013, S. 1)

4.3.1.2 Mini-Languages

Brusilovsky, Calabrese, Hvorecky, Kouchnirenko & Miller (1997) präsentieren einen historischen Abriss über „Mini-Languages“, einfache visuelle sowie intuitive Programmiersprachen zum Erlernen der Programmierung, welche auf Paperts LOGO und seiner „Turtle-Graphics“ basieren. Bestehende Programmiersprachen haben (1997) einen zu großen Funktionsumfang, sind nicht visuell und grundlegende Strukturen sind unter komplexen Konstrukten verborgen sowie die verwendeten Symbole und Elemente sind den Anfängerinnen und Anfängern nicht vertraut. Um interessante Programme entwickeln zu können ist ein vertiefendes Verständnis der jeweiligen Programmiersprache von Nöten, was den Einstieg erschwert. (Brusilovsky et al., 1997, S. 2 f.)

Das Ziel einer „Mini-Language“ ist es, eine einfache und wenige Befehle umfassende Programmiersprache zu entwickeln, die zur Unterstützung beim Erlernen von Programmiersprachen dienen kann. Gemeinsam ist den bereits bestehenden „Mini-Languages“, dass Lernende Programmieren durch die Aneignung der Steuerung eines Akteurs (Objekts) erlernen. Eine „Mini-Language“ muss, so Brusilovsky et al. (1997), in Bezug auf Semantik und Syntax, einfach gestaltet sein. Darüber hinaus müssen jegliche durchgeführten Operationen mit den Akteurinnen und Akteuren durch eine Veränderung der Mikrowelt in Annäherung an die Realität dargestellt werden. Attraktiv und bedeutungsvoll sollen die „Mini-Languages“ ebenso sein, wie unterhaltend. Ebenfalls sollte eine Modularität durch die Möglichkeit zur Gestaltung eigener Prozeduren gegeben sein. (Brusilovsky et al., 1997, S. 7) Diese Kriterien sind ähnlich zu den allgemeinen Anforderungen an die Usability von Software (siehe dazu auch Nielsen (1993) und Kapitel 5.1.2).

Die „Mini-Languages“ sollen die Möglichkeit für das systematische Lösen von Problemen / Aufgabenstellungen bieten und gleichzeitig das algorithmische Denken fördern. Die Gründe, die das Erlernen einer „Mini-Language“ bereits im Schulalter (von Programmieren im Kindergarten sprach 1997 noch kaum jemand) attraktiv erscheinen lassen, sind nach Brusilovsky et al. (1997) nicht nur interessant, sondern bilden auch die Grundlage für zahlreiche weitere Überlegungen:

- Die Programmierung von Computern bildet die Basis für logisches und abstraktes Denken, welches als fundamental für den Lernprozess erachtet wird.
- „Mini-Languages“ sind in ihrem Funktionsumfang klein gehalten und erlauben in Folge ein schnelles Erlernen. Das führt zu raschen Erfolgserlebnissen, was wiederum der Motivation der Lernenden dienlich ist.
- Die schnelle Einarbeitung in „Mini-Languages“ bietet Gelegenheit, sich auf weit wichtigere Themen, als die Programmiersprache selbst zu konzentrieren: die Entwicklung von Algorithmen, das Design von Software oder grundlegende Programmierkonzepte.
- „Mini-Languages“ basieren auf Metaphern aus dem Alltag – was förderlich für die intrinsische Motivation der Lernenden ist. „It is possible to create rich sets of

problems that both cover the fundamental ideas and dovetail with the students life experiences.“ (Brusilovsky et al., 1997, S. 3)

- Der direkte Zusammenhang zwischen der Semantik der Sprachkonstrukte und den (Re)Aktionen der zum Einsatz kommenden Akteurinnen und Akteure, erleichtert das Verständnis für die Materie des Programmierens.
- „Mini-Languages“ sind überwiegend für Lehrzwecke entwickelt worden – und werden nach wie vor vorwiegend für selbige entwickelt.
„[...] a mini-language can take advantage of the narrow definition of the class of its users by using students' native language for keywords and providing only data types and control structures essential for these students.“ (Brusilovsky et al., 1997, S. 3)

(Brusilovsky et al., 1997, S. 3)

„Mini-Languages“ erscheinen daher ideal geeignet für einen Einsatz mit Programmieranfängerinnen und -anfängern, da sie sich mit einem minimalen Befehlsset begnügen und so einen niederschweligen Einstieg ermöglichen.

4.3.1.3 Exemplarische Programmier-Lernumgebungen für die Zielgruppe der 3- bis 6-jährigen Kinder

Ramírez-Benavides & Guerrero (2015) erstellten eine Übersicht über das Angebot an Programmierumgebungen und klassifizierten dieses nach Altersstufen der angedachten Zielgruppen. (Ramírez-Benavides & Guerrero, 2015, S. 154) Im Folgenden werden Programmierumgebungen / Mikrowelten vorgestellt, die für die Zielgruppe der 3- bis 6-jährigen Kinder noch am ehesten in Frage kommen könnten.

4.3.1.3.1 *ToonTalk, Kidsim & Cocoa, Stagecast Creator*

ToonTalk ist eine Programmiersprache, deren Quellcode animiert ist und so die Programmierung als Ergebnis zwischen dem Wunsch einer Befehlsausführung und der tatsächlichen Ausführung des Befehls durch den Computer visualisiert. ToonTalk ist gemäß den Autoren für Kindergartenkinder geeignet, erfordert allerdings nach Ansicht der Verfasserin der gegenständlichen Arbeit erhebliche Lesefähigkeiten für die Programmierung, welche durch ein tutorielles System unterstützt wird. (Kahn, 1996), (Kahn, 1996b), (Morgado & Kahn, 2008) sowie (Kahn, 2014)

Erwähnenswert in Hinblick auf die eingesetzten Konzepte sind Kidsim und später Cocoa (die Weiterentwicklung von Kidsim), die vor allem durch das Konzept der Programmierung durch Demonstration (PBD – Programming by demonstration) und den Einsatz von grafischen „Rewrite-Rules“ den Grundstein für viele heutige Programmierumgebungen für 3- bis 6-jährige Kinder gelegt haben. (Smith, Cypher & Schmucker, 1996), (Smith, 1994), (Cypher & Smith, 1995), (Rader, Brand & Lewis, 1997), (Green & Petre, 1996), (Kindborg & McGee, 2007) sowie (Xiajian, Danli & Hongan, 2011)

Die ständige Forschung und Weiterentwicklung von Cocoa brachte in weiterer Folge das bekannte und auch kommerziell erhältliche Produkt Stagecast Creator (das allerdings erst für Kinder ab 8 Jahren geeignet ist) hervor, das ebenfalls auf PBD und grafischen „Rewrite-Rules“ basiert. (Smith, Cypher & Tesler, 2000), (Wright, 2006) Ein ähnliches Produkt ist Leogos. (Cockburn & Bryant, 1998) Darüber hinaus kommen auch Modelleisenbahnen zur Vermittlung grundlegender Konzepte der Programmierung zum Einsatz. (Noma, Sasamoto, Itoh, Kitamura, Kishino & Tetsutani, 2003).

4.3.1.3.2 *ScratchJr*

Sehr populär sind unter anderem auch Scratch und ScratchJr. Scratch ist eine visuelle Programmiersprache, die für Kinder im Alter von 8 bis 16 Jahren geeignet ist. (Maloney, Resnick, Rusk, Silverman & Eastmond, 2010, S. 1) Scratch basiert auf den Ideen von LOGO, ersetzt jedoch den zu schreibenden Programmcode durch Elemente, die per Drag & Drop aneinandergereiht und zu einem Programm zusammengeführt werden können. In Scratch können Medienelemente manipuliert und so die Interessen von Kindern, wie die

Erstellung von Animationen, Spielen oder interaktiven Präsentationen, unterstützt werden. (Maloney, Peppler, Kafai, Resnick & Rusk, 2008, S. 367), (Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, Millner, Rosenbaum, Silver, Silverman & Kafai, 2009)

In einer Weiterentwicklung ist ScratchJr (Scratch Junior) für Kindergartenkinder bis zu einem Alter von 7 Jahren entstanden. ScratchJr verfügt über eine für die Altersgruppe geeignete grafische Oberfläche, begleitende Lehr-Lern-Materialien, integrierte Übungen für Mathematik, zum Schreibenlernen sowie für andere kognitive Fähigkeiten. Eine Online-Community steht Kindergartenpädagoginnen und -pädagogen zum gegenseitigen Erfahrungsaustausch und zur Materialiensammlung zur Verfügung. (Flannery et al., 2013, S. 1) Der Erfolg von ScratchJr führt dazu, dass die Entwicklerinnen und Entwickler weitere Aktivitäten und Materialien rund um ScratchJr zur Verfügung stellen und so für eine stärkere Verankerung in Curricula und schulischen Aktivitäten sorgen. (Strawhacker, Lee, Caine & Bers, 2015, S. 3)

4.3.1.3.3 Focus on Bee-Bot

Focus on Bee-Bot¹¹ stellt eine graphische Simulation des Bee-Bot Bodenroboters zur Verfügung. Die Software wurde für Vor- und Volksschulkinder konzipiert und orientiert sich am Lehrplan der Volksschule in Großbritannien. Um die Software mit ihrem ganzen Umfang eigenständig nutzen zu können, müssen die Kinder einerseits lesen können und andererseits englischsprachig sein. Eine grundlegende Bedienung ist jedoch auch ohne Schreib- und Lesefähigkeiten möglich.

Sowohl bei der Arbeit mit dem realen Roboter als auch bei der Arbeit mit dem Lernprogramm Focus on Bee-Bot lernen die Kinder die Steuerungs- und Richtungssprache des Roboters einzusetzen und eine Abfolge (Sequenz) von Befehlen zu programmieren. Dies ist aufgrund der Symbole auf den Tasten auch für Kinder möglich, die noch nicht lesen und schreiben können.

Grundsätzlich muss der Bee-Bot am Bildschirm über sog. Matten navigiert werden, um von Erwachsenen oder vom Kind selbst gestellte Aufgaben erfolgreich zu absolvieren. Dies entspricht auch dem realen Bodenroboter Bee-Bot, der durch individuelle Programmierung von den Kindern gesteuert wird.

4.3.1.4 Exemplarische Programmier-Lernumgebungen für ältere Kinder

Für Kinder älter als 6 Jahre sind zahlreiche, zum Teil sehr spezialisierte, Lernumgebungen verfügbar, wie z.B. Magic Words, wo Kinder mit Hilfe von Wortkarten nicht nur Programmieren, sondern auch Lesen und Fremdsprachen lernen können. (Kindborg & Sökjer, 2007) Mit AgentCubes können Jugendliche ab 10 Jahren 3D Spiele programmieren (Ioannidou, Repenning & Webb, 2009) und für interaktive Simulationen unterschiedlicher Fachdisziplinen steht AgentSheets zur Verfügung (Repenning, 2000).

4.3.2 Tangible Interfaces

Angesehene Pädagoginnen und Pädagogen wie Friedrich Fröbel, Maria Montessori, Célestin Freinet oder Jean Piaget waren der Ansicht, dass Lernen durch Angreifen und den Einsatz von physischen Objekten bevorzugt werden sollte. Das Wissen über die Lerngewohnheiten der Kinder kann dazu verwendet werden, logische und abstrakte Denkweisen sowie erste Programmiererfahrungen zu fördern. (Scharf, Winkler & Herczeg, 2008, S. 243)

„Tangible interfaces are of special interest in early childhood education as they resonate with traditional learning manipulates, such as Montessori blocks and Froebel’s gifts specifically designed to learn about mathematical concepts [...]“ (Strawhacker & Bers, 2014, S. 293 f.)

¹¹ <https://www.focuseducational.com/product/bee-bot-lesson-activities-2/96>, 23.04.2018

Nach Aussage von Scharf, Winkler & Herczeg (2008) fühlen sich Kindergartenkinder von Computern überfordert, da hierbei die Technologie im Zentrum der Aufmerksamkeit liegt. Computer werden dadurch für Kinder schwer zu bedienen und zu beherrschen. Auch die enthaltene Funktionsvielfalt in Standardprogrammen kann leicht überfordernd werden. (Scharf, Winkler & Herczeg, 2008, S. 243) Je jünger die Zielgruppe ist, desto geringer sollten die Abstraktion und die Komplexität der verwendeten Materialien und Technologien sein, denn auch mit vereinfachten Strukturen können die Grundlagen für abstraktere und komplexere Aufgaben und deren Lösungen erarbeitet werden. (Scharf, Winkler & Herczeg, 2008, S. 244)

Gerade auch die weit verbreiteten und bekannten LEGO® Mindstorms® sind für junge Kinder schwer zu verstehen, denn die Programmierung von Mikrocontrollern am PC mit anschließender Datenübertragung auf den Roboter ist für Kindergartenkinder zu komplex und abstrakt, darüber hinaus ist der Funktionsumfang viel zu groß. (Scharf, Winkler & Herczeg, 2008, S. 244 f.)

Bers & Horn (2010) definieren eine Tangible Programming Language, also eine angreifbare Programmiersprache, wie folgt:

„A tangible programming language, like any other type of computer language, is simply a tool for telling a computer what to do. [...] Instead of relying on pictures and words on a computer screen, tangible languages use physical objects to represent the various aspects of computer programming. Users arrange and connect these physical elements to construct programs. Rather than falling back on implied rules and conventions, tangible languages can exploit the physical properties of objects, such as size, shape, and material to express and enforce syntax. [...] In fact, with this language, while it is possible to make mistakes in program logic, it is impossible to produce a syntax error.” (Bers & Horn, 2010, S. 55)

Physische Objekte repräsentieren unterschiedliche Aspekte von Programmiersprachen – vor allem die einzelnen Befehle, die in einer Programmiersprache aneinandergereiht werden können. Der größte Vorteil in der Nutzung von physikalischen bzw. angreifbaren Objekten liegt darin, dass man zwar Fehler in der Reihenfolge der Objekte, und damit in der Programmlogik, machen kann, sich aber nicht mit Syntax-Fehlern beschäftigen muss, die die komplette Ausführung eines Programms verhindern können. Somit ist in Folge die Programmierung einfacher erlernbar und man kann sich auf den Inhalt der Programmierung und die Tätigkeit des Programmierens konzentrieren. Die Idee des angreifbaren Programmierens geht in die Mitte der 1970er Jahre auf Radia Perlman zurück. (Bers & Horn, 2010, S. 55)

„The presence of several types of physical computing and robotics devices in educational settings is attributable to the many research and design initiatives of the past thirty years. [...] the design of such devices has evolved significantly, and their popularity has grown significantly [...].” (Blikstein, 2013, S. 173)

Tangible Interfaces bilden aber auch die Grundlage für die Erforschung von neuen Interaktionsmöglichkeiten mit dem Computer. Die natürliche und direkte Manipulation von Objekten erleichtert die Interaktion mit einem Computer sehr und erlaubt, dass sich Menschen ein eigenes Regelwerk zur Interaktion mit dem Computer zurechtlegen können. (Montemayor, 2001, S. 81) Montemayor (2001) spricht in diesem Zusammenhang von Physical Programming Languages, die durch Interaktion mit physischen Objekten eine Programmierung selbiger herbeiführen. (Montemayor, 2001, S. 81 f.)

Für Kinder im Alter von 5 bis 13 Jahren eignet sich das roBlocks-System (das in verschiedenen Evolutionsstufen unter den Namen Modular Robots, roBlocks und Cubelets bekannt ist), das aus Robotikblöcken und einem Softwarepaket besteht, das Benutzerinnen und Benutzern ermöglicht, durch Verbinden der Blöcke zu programmieren, ihre Konstruktionen durch Bildschirm-Tools zu analysieren und schließlich ihre eigenen Kreationen neu zu programmieren. (Schweikardt & Gross, 2006), (Schweikardt & Gross, 2008)

Durchwegs für ältere Kinder sind unter anderem folgende Tangible Interfaces geeignet:

- Paper Computing und Ambient Programming (Eisenberg, Elumeze, MacFerrin & Buechley, 2009),
- Programming on a Planetarium Sphere (Eisenberg et al., 2009),
- Topobo (Raffle, Parkes & Ishii, 2004),
- ActiveCubes und PlayCubes (Jacoby, Gutwillig, Jacoby, Josman, Weiss, Koike, Itoh, Kawai, Kitamura & Sharlin, 2009),
- SmartTiles (Elumeze & Eisenberg, 2005),
- Digital Construction Set (McNerney, 2004),
- Kinematics (Oschuetz, Wessolek & Sattler, 2010),
- Tangible Cubes as Programming Objects (Smith, 2006),
- Quetzal (Horn & Jacob, 2007b),
- T-Maze (Wang, Zhang & Wang, 2011),
- Birdwatcher (Elumeze, Huang, Meyers & Eisenberg, 2010),
- MagicPlayroom (Iwata, Tano, Fujie & Hashiyama, 2006),
- UbiPlay (Mattila & Vääänen, 2006).

4.3.2.1 Tangicons

Speziell für den Einsatz im Kindergarten und für junge Schulkinder wurden von Scharf, Winkler & Herczeg (2008) die sogenannten Tangicons entwickelt. Hierbei handelt es sich um nichtelektronische, physische Programmierwürfel, die gemeinsam mit Kindergartenkindern in mehreren Einheiten entwickelt wurden, wobei der Entwicklungsprozess nicht näher beschrieben ist. (Scharf, Winkler & Herczeg, 2008, S. 242)

Der Bedarf zur Entwicklung der Tangicons ergab sich nach Ansicht der Entwicklerinnen und Entwickler aus der Tatsache, dass die meisten am Markt befindlichen Umgebungen mit Tangible Interfaces zu komplex für die Zielgruppe waren. Die angreifbaren, programmierbaren Blöcke, mit denen Sequenzen von Handlungen programmiert werden können, wurden basierend auf pädagogischen Grundlagen entwickelt. Die Gestaltung der verwendeten Symbole bezieht sich hierbei auf die reale Welt, um den Kindern in einer vertrauten Umgebung und auf spielerische Weise die ersten Schritte der Programmierung zu vermitteln. (Scharf, Winkler & Herczeg, 2008, S. 242)

Scharf, Winkler & Herczeg (2008) identifizierten bei Michael Horns Programmiersprache Quetzal (Horn & Jacob, 2007b) die komplexen Symbole als Problem für jüngere Kinder. Die angreifbaren Blöcke repräsentieren hier exakt die visuelle Version der Quetzal Programmierumgebung, die sehr ähnlich zur Programmierung mit Robolab¹² ist. Junge Kinder haben allerdings Probleme diese abstrakten Objekte zu entschlüsseln, da diese keine Querverbindung zu den Objekten in ihrem täglichen Leben haben. Sie benötigen angreifbare Objekte, die auf einen Blick erfasst und interpretiert werden können. (Scharf, Winkler & Herczeg, 2008, S. 245 f.)

Waren die Tangicons in den ersten beiden Versionen noch für Kinder im Alter von 4 bis 6 Jahren geeignet, ist die dritte Version primär für 6- bis 9-jährige Kinder, da das Spiel, welches mit den Tangicons gespielt werden kann, umfangreicher wurde und auch die Möglichkeiten der Tangicons selbst vielfältiger wurden. (Scharf, Winkler, Hahn, Wolters & Herczeg, 2012, S. 145) Durch das Spielen erhalten die Kinder einen natürlichen Zugang zu Funktionen, Parametern und Sequenzen. (Scharf et al., 2012, S. 150)

¹² Robolab ist eine grafische Benutzerschnittstelle zur Programmierung eines RCX (siehe Kapitel 4.4.2). (Bers, Rogers, Beals, Portsmore, Staszowski, Cejka, Carberry, Gravel, Anderson & Barnett, 2006, S. 1036)

4.3.2.2 Electronic Blocks

Wyeth & Purchase (2000, 2002, 2002b) entwickelten die sogenannten „Electronic Blocks“, ein System zum Programmieren lernen für Kinder zwischen 3 und 8 Jahren. Dabei handelt es sich um kleine Quader, die mit einfachen elektronischen Elementen ausgestattet sind und Kindern die Möglichkeit bieten, durch Kombinieren und Zusammenstecken der Blöcke, neue, komplexere Elemente und Systeme zu erschaffen. (Wyeth & Purchase, 2002b) Bei der Entwicklung der Electronic Blocks wurde nicht nur auf Design-Aspekte viel Wert gelegt, sondern auch auf eine zielgruppengerechte, pädagogisch passende Lösung. (Wyeth & Purchase, 2002, S. 227)

Im Rahmen einer Evaluation wurden 28 Vorschulkinder bei der Nutzung von Electronic Blocks beobachtet. Alle beobachteten Kinder bauten ein Auto, welches mit Sensoren gesteuert werden konnte. Die Vielzahl der Kombinationsmöglichkeiten der Blöcke erlaubte einigen Kindern per Zufall auf die Konstruktion eines Autos zu stoßen, andere wiederum bauten dieses bewusst. (Wyeth & Purchase, 2002, S. 235) Die Komplexität der gebauten Konstrukte variierte stark. Von einfachen aus zwei Blöcken bestehenden Systemen bis hin zu großen Türmen. (Wyeth & Purchase, 2002, S. 235 & Wyeth & Weyth, 2001, S. 502) Kinder trainierten bei der Beschäftigung mit den Electronic Blocks programmertypische Aspekte, wie die Wiederverwendung von Code (durch die erneute Nutzung bereits zusammengesteckter Elemente), Debugging (Nutzung des Trial & Error-Prinzips um Fehler zu finden) und die Kombination von Befehlen (Zusammensetzen von Elementen, die in ihrer Kombination korrekt funktionieren). (Wyeth & Weyth, 2001, S. 502)

4.3.2.3 Kiwi, Kibo, Tern und Cherp

Neben der wachsenden Anzahl an Berichten über die Wirksamkeit und den Nutzen eines Einsatzes von Robotern in Schulen, zeigt sich, dass es nur sehr wenige Roboter und Robotersets am Markt gibt, die speziell für den Einsatz im Kindergarten und für den Schulbeginn geeignet sind und sowohl Gestaltungs- als auch Programmierfertigkeiten fördern. (Sullivan & Bers, 2015, S. 3)

Sullivan & Bers (2015) berichten in Folge über die Entwicklung von KIWI, einem Roboter, der mittels CHERP (Creative Hybrid Environment for Robotic Programming) programmiert werden kann. Bei der Entwicklung lag der Fokus auf der Zielgruppe der 4- bis 7-jährigen Kinder. (Sullivan & Bers, 2015, S. 4)

Als Basis für CHERP dient die Programmiersprache Tern. In Tern werden Befehle, Schleifen, Verzweigungen und Unterprogramme durch die Aneinanderreihung von hölzernen Bauklötzen in Form von Puzzle-Steinen repräsentiert. (Horn & Jacob, 2007, S. 1965) Tern verwendet die text-basierte Programmiersprache Karel the Robot¹³, die Pascal-ähnliche Programme erlaubt und keine Variablen, keine Parameterwerte und nur ein kleines Set an Befehlen anbietet. (Horn & Jacob, 2007, S. 1966)

CHERP besteht aus farbkodierten Programmierbefehlen, wie ein Schritt vorwärts oder Beep und kann entweder durch Drag & Drop Aktionen am Bildschirm, zur Aneinanderreihung der Symbole in eine Sequenz, bedient werden oder durch Aneinanderreihung von Holzblöcken, die die notwendigen Symbole aufgedruckt haben. (Strawhacker & Bers, 2014, S. 298) Die Programmierung durch die Holzblöcke wird am Bildschirm repräsentiert und lässt den Kindern die Möglichkeit zu wählen, mit welcher Variante sie programmieren möchten. (Kazakoff & Bers, 2012, S. 379), (Horn, Crouser & Bers, 2012, S. 379)

Der Roboter KIWI kann mit einem Scanner die Barcodes auf den Holzblöcken, die selbst über keine Elektronik verfügen, lesen und so ohne Computer programmiert werden. (Sullivan & Bers, 2015, S. 5) Während KIWI noch ein Prototyp war, der in einer dreijährigen

¹³ <http://karel.sourceforge.net/>, 23.04.2018

Entwicklung verfeinert und weiterentwickelt wurde, ist nun KIBO das Produkt, welches am freien Markt erhältlich ist.¹⁴ (Sullivan, Elkin & Bers, 2015, S. 418)

4.3.2.4 Cubetto und ROBOT Turtles

Für Kinder ab 3 Jahren steht Cubetto zur Verfügung. Mit Cubetto können erste Programmiererfahrungen gesammelt werden. Auf einem Brettspiel können die Programmierbefehle angeordnet werden, die anschließend an den Cubetto übertragen werden, der die Befehle ausführt. Der Roboter und die angreifbaren Elemente sind vorrangig aus Holz gefertigt.¹⁵ Das Brettspiel ROBOT Turtles vermittelt in spielerischer Form 3- bis 8-Jährigen die ersten Schritte der Programmierung. Zur Ergänzung steht ein interaktives E-Book zur Verfügung.¹⁶

4.3.3 **Tangible Interfaces vs. Grafische Benutzeroberflächen**

Horn, Solovey, Crouser & Jacob (2009) führten eine Vergleichsstudie durch, um herauszufinden, ob von Programmieranfängerinnen und -anfängern eher Tangible Interfaces oder grafische Benutzeroberflächen genutzt werden um in die Welt der Programmierung einzusteigen. Dazu wurde im Rahmen einer Ausstellung in Boston eine Programmierumgebung sowohl in Form eines Tangible Interfaces als auch in Form einer grafischen Benutzeroberfläche als Software zur Verfügung gestellt. (Horn et al., 2009, S. 975) Das Tangible Interface bestand aus hölzernen Puzzlesteinen, die in einer beliebigen Reihenfolge aneinandergereiht werden konnten – jeder von ihnen repräsentierte einen Befehl, wie Schütteln, Piep, Linksdrehen usw. Auch einfache Schleifen, Wiederholungen, konnten programmiert werden. (Horn, Solovey & Jacob, 2008, S. 197) Die Aneinanderreihung der Puzzlesteine wurde von einer Bilderkennung erfasst, durch eine eigenentwickelte Software in Programmcode konvertiert und an einen Bodenroboter der Marke iRobot Create™ via Bluetooth übertragen. (Horn et al., 2009, S. 976 f.) Für die graphische Benutzeroberfläche wurde an einem Rechner die Programmierumgebung Scratch (siehe Kapitel 4.3.1.3.2) eingesetzt. Das fertige Programm wurde wiederum von der eigenentwickelten Software in Programmcode übersetzt und an den iRobot Create™ via Bluetooth übertragen. (Horn et al., 2009, S. 976 f.)

Im Rahmen der genannten Ausstellung wurden die teilnehmenden Personen, wie Kinder, Familien und Gruppen beobachtet und befragt. Die überwiegende Mehrheit der Probandinnen und Probanden entschied sich für die Nutzung des Tangible Interfaces. (Horn et al., 2009, S. 978) Vor allem für Kinder und hier besonders für Mädchen geht aus den Beobachtungen eindeutig hervor, dass sie lieber das Tangible Interface nutzen. Auch wird klar erkennbar, dass deutlich intensiver zusammengearbeitet wird, wenn Tangible Interfaces genutzt werden, als wenn mehrere Personen rund um einen Computer-Bildschirm stehen und versuchen mit Hilfe einer grafischen Benutzeroberfläche Probleme zu lösen. Die physischen Objekte erlauben hier eine gemeinsame und dennoch auch individuelle Lösung zu erarbeiten, da jede und jeder Beteiligte eigene Elemente manipulieren und zu einem Ganzen zusammenführen kann. (Horn et al., 2009, S. 980) Es zeigt sich auch, dass die Lösungen, die mit Hilfe des Tangible Interfaces entwickelt wurden, deutlich längere und komplexere Aktionen des Roboters mit sich brachten. (Horn et al., 2009, S. 981) Bei Familien konnte beobachtet werden, dass bei der Nutzung von Tangible Interfaces die Kinder deutlich mehr Interesse und Beteiligung zeigten, als bei der Nutzung der grafischen Benutzeroberfläche. (Horn et al., 2009, S. 981) Die Autorinnen und Autoren fassen die Ergebnisse ihrer Beobachtungen wie folgt zusammen:

¹⁴ <http://www.kinderlabrobotics.com>, 23.04.2018

¹⁵ <https://www.primotoys.com/>, 23.04.2018

¹⁶ <http://www.robotturtles.com/>, 23.04.2018

„Our results provide concrete evidence that thoughtfully designed tangible interfaces can offer several significant advantages over the standard single-mouse graphical interfaces in the context of informal science education. Among these advantages, tangible interfaces can be more inviting and more conducive to collaborative interaction. Furthermore, in this case, the tangible interface is better at encouraging children to take an active role in exploring and learning, an effect that seems especially strong for girls.” (Horn et al., 2009, S. 982)

Ähnliches wird von Brauner et al. (2010) berichtet, wenn auch nicht so eindeutig. Jedoch zeigt sich auch dort, dass Tangible Interfaces eher für Programmieranfängerinnen und -anfänger geeignet sind und die Lösungen von höherer Qualität sind, als bei Einsatz einer grafischen Benutzeroberfläche. (Brauner et al., 2010, S. 67)

Dass die visuelle Repräsentation bzw. die grafische Benutzeroberfläche ungeahnte Hürden in der Bedienung mit sich bringen kann, zeigen Verhaegh, Fontijn & Jacobs (2008) auf, die zum Schluss kommen, dass für Kinder Tangible Interfaces nicht nur leichter zu erlernen und zu nutzen sind, sondern dass sie durch die einfache Bedienung weniger vom eigentlichen Gegenstand – der Programmierung – abgelenkt werden. Wohingegen die grafische Oberfläche deutlich mehr Aufmerksamkeit in der Bedienung erfordert. (Verhaegh, Fontijn & Jacobs, 2008, S. 145) Auch sind sie der Ansicht, dass die Benutzerschnittstelle nicht zu einer zusätzlichen Herausforderung werden darf (Verhaegh, Fontijn & Jacobs, 2008, S. 141) und Tangible Interfaces die Experimentierfreude der Kinder erhöhen. (Verhaegh, Fontijn & Jacobs, 2008, S. 145), (Smith, 2009, S. 288 f.) Darüber hinaus sind Tangible Interfaces für Kinder vertrauter, direkter begreifbar und steigern das Interesse zur Benutzung. (Xiajian, Danli & Hongan, 2011, S. 572) Der Einsatz von derartigen Technologien birgt das Potenzial in sich, die Phantasie von Kindern anzuregen. (Mansor, De Angeli & De Bruin, 2008, S. 93)

Wu, Tseng & Huang (2008) führten mit älteren Kindern ebenfalls einen Vergleich zwischen Tangible Interfaces und einer grafischen Benutzeroberfläche durch, wobei der Fokus nicht auf der Programmierung, sondern in der Ausführung lag. Die Programmierung wurde entweder auf einem realen LEGO® Mindstorm®-Roboter oder auf einem LEGO® Mindstorm®-Simulator am Bildschirm ausgeführt. (Wu, Tseng & Huang, 2008, S. 53) Auch bei älteren Kindern zeigte sich, dass diese Reaktionen einzelner Befehle am realen Objekt eher verstehen und begreifen, als wenn sie es nur am Bildschirm sehen. Obwohl das Ergebnis in Form einer Simulation am Bildschirm eher verfügbar ist als die Präsentation am realen Objekt, weil dabei als Zwischenschritt zuerst das fertige Programm auf den realen Roboter geladen werden muss, (Wu, Tseng & Huang, 2008, S. 60) war die Motivation weitere Programme zu schreiben und Experimente durchzuführen trotz des höheren Aufwands bei der Gruppe an älteren Kindern höher, die die Ergebnisse am realen Objekt wahrnehmen konnten. (Wu, Tseng & Huang, 2008, S. 61)

Marshall (2007) hinterfragt in seiner Arbeit die Vorteile von Tangible Interfaces kritisch. Er sieht hier einen geringen bis gar keinen Vorteil im Einsatz derselben – im Gegenteil, softwaregestützte Materialien sind aus seiner Sicht flexibler und einfacher zum Einsatz zu bringen. (Marshall, 2007, S. 168 f.) Hier ist jedoch anzumerken, dass sich Marshalls Arbeit nicht auf das Erlernen von Programmierung bezieht, sondern auf den Einsatz von Tangible Interfaces allgemein. Auch ortet er noch einen erheblichen Forschungsbedarf um den Mehrwert bzw. den Nutzen von Tangible Interfaces zu bestätigen. (Marshall, 2007, S. 168 f.)

4.4 Educational Robotics

Bei Educational Robotics handelt es sich um ein junges Forschungsgebiet, das sich mit dem Einsatz, der Gestaltung und Evaluierung von Robotern in der Lehre, im Speziellen auch im Kindergarten, in der Volksschule und Sekundarstufe beschäftigt.¹⁷ Aber auch die Nutzung in der Hochschullehre fällt durchaus in dieses Forschungsgebiet. Im Folgenden wird das

¹⁷ Eine genauere Differenzierung der Terminologien in diesem Bereich ist u.a. Angel-Fernandez & Vincze (2018b) zu entnehmen.

Themengebiet „Robotics in Education“ vorgestellt und umrissen, sowie exemplarische Bodenroboter vorgestellt. Eine mögliche Forschungslücke in diesem Themengebiet und im Kontext der gegenständlichen Arbeit ist in Kapitel 6.2 dargelegt.

4.4.1 Robotics in Education

Robotics in Education beschreibt den Einsatz von Robotern in der Lehre und ist seit ca. 20 Jahren in der Literatur als eigenständiges Forschungsgebiet wahrzunehmen. Eguchi (2007) beschreibt die Anfänge von Robotics in Education. Die Popularität steigerte sich mit zunehmender Verfügbarkeit von geeigneten Robotern und den Möglichkeiten an Wettbewerben teilzunehmen, um sich miteinander zu messen. Einer der wesentlichen Player in diesem Segment war und ist nach wie vor die FIRST – „For Inspiration und Recognition of Science and Technology“ – Vereinigung, die als Non-Profit-Organisation jährliche nationale und internationale Wettbewerbe, ausgehend von den USA, veranstaltet. Neben der klassischen FIRST-Liga gibt es auch noch die FIRST-LEGO®-Liga, in der mit LEGO® Mindstorms®-Robotern 9- bis 14-jährige Kinder untereinander wetteifern. (Eguchi, 2007, S. 2542) All diese (Lehr-)Aktivitäten fasst Eguchi (2009) unter dem Titel „Goal-Oriented Approach“, also zielorientierter Ansatz zusammen. (Eguchi, 2009, S. 3548)

Forschungsergebnisse für die Altersgruppe der 7- bis 10-jährigen Kinder sind unter anderem bei (Eguchi & Uribe, 2009), (Goh & Aris, 2007), (Chan, Black, Han, Vitale, Xia, Subramanian, Du & Kang, 2007), (Eguchi, Shou & Shen, 2010) oder (Hirst, Johnson, Petre, Price & Richards, 2003) nachzulesen.

Bers & Horn (2010) sind überzeugt, dass Roboter einen großen Mehrwert in der Lehre / Bildung, speziell für junge Kinder, darstellen und sowohl kognitive, motorische als auch soziale Kompetenzen fördern können. (Bers & Horn, 2010, S. 51) Der Einsatz von Robotern bietet die Chance und Möglichkeiten dafür, dass junge Kinder ein Verständnis über Sensoren, Motoren und die digitale Welt spielerisch erwerben können. Die Kinder schlüpfen in die Rollen von Technikerinnen und Technikern, von Gestalterinnen und Gestaltern während sie mit Zahnrädern, Hebeln, Motoren, Sensoren und Programmier-Schleifen spielen und mit Geschichten ihre eigenen Projekte bereichern – das alles in ihrem eigenen Kontext, in ihrer persönlichen Umwelt. (Bers & Horn, 2010, S. 52) Bereits 4-jährige Kinder sind in der Lage, grundlegende Computerprogramme zu verstehen und auf Roboter anzuwenden. (Bers & Horn, 2010, S. 52)

Barker, Grandgenett, Nugent & Adamchuk (2009) zeigen auf, dass die Beschäftigung mit Robotern durchwegs einen positiven Einfluss auf die STEM-Fächer¹⁸ hat. Vor allem eine fächerübergreifende Beschäftigung mit Robotern und das Schaffen von Lernräumen, in denen informell gelernt werden kann, erscheinen ihnen sehr wichtig. (Barker et al., 2009, S. 3235 f.) Die positiven Auswirkungen auf das (natur-)wissenschaftliche Verständnis, welches im Rahmen der Robotik-Aktivitäten erworben wird, heben auch Li, Han, Kang, Lu & Black (2009) hervor. (Li et al., 2009, S. 2960) Dies unterstreicht auch Johnson (2003), der Robotik als einen multidisziplinären Ansatz zum Lernen ansieht, der sehr motivierend sein kann. Darüber hinaus war sich Johnson (2003) schon 2003 sicher, dass Robotik nicht nur als „Hype“ zu betrachten ist, sondern auf lange Zeit eine wesentliche Rolle im Bildungswesen spielen wird. (Johnson, 2003, S. 20) Bell, Heeler & Heeler (2008) verweisen zusätzlich auf die hohe miteinhergehende Motivation zu lernen.

Fernaesus, Håkansson, Jacobsson & Ljungblad (2010) beschäftigen sich in ihrer Forschung mit Roboterspielzeugen, welche sie als Roboter definieren, die für Freizeitaktivitäten wie Spielen, kreatives Gestalten, spielerisches Lernen, Unterhaltung und Entspannung entwickelt wurden. Diese Spielzeuge enthalten eine Softwarekomponente, die sie von anderen mechanischen oder weniger technischen Artefakten unterscheidet. Sie interagieren

¹⁸ STEM - Science, Technology, Engineering and Mathematics bzw. im deutschsprachigen Raum MINT-Fächer – Mathematik, Informatik, Naturwissenschaften und Technik.

direkt mit ihrer Umwelt, im Gegensatz zu Software, die am Computer oder Mobilgerät ausgeführt wird. (Fernaues et al., 2010, S. 39) Beispielhaft führen Fernaeus et al. (2010) Pleo als Roboterspielzeug an, welches auch mit einer eigenen grafischen Benutzeroberfläche genutzt werden kann. (Ryokai, Lee & Breitbart, 2009) Die Stärke derartiger Spielzeug-Roboter liegt u.a. auch in einer Möglichkeit, schon sehr junge Kinder mit Robotern in Berührung zu bringen.

Hyun & Yoon (2009) erforschten das Verhalten junger Kinder beim Einsatz eines Spielzeug-Roboters. Sie beobachteten drei Monate lang 3- bis 4-jährige Kinder in Kindergärten, die in ihrer freien Spielzeit mit einem Roboter des Typs iRobiQ¹⁹ spielen durften. Über die Funktionen des Roboters selbst oder die möglichen Einsatzzwecke wurde zu Beginn nicht gesprochen. (Hyun & Yoon, 2009, S. 677) Hyun & Yoon (2009) stellten fest, dass Kinder sich interessierter an Robotern und der Interaktion mit ihnen zeigen, je jünger sie sind. (Hyun & Yoon, 2009, S. 675) Die Dauer der Nutzung war ihren Beobachtungen nach direkt proportional zu deren Häufigkeit. Darüber hinaus wurde der Abstand des Kindes zum Roboter immer geringer, je öfter das Kind mit dem Roboter spielte. Die Bedienung erfolgte meistens zu zweit, wobei beobachtet wurde, dass immer ein Kind das „führende“ Kind war, welches die Befehle über das Display am Roboter aufrief sowie die Aktionen mit dem Roboter startete und das andere Kind sowohl assistierte als auch an der Diskussion mitwirkte. Manche Kinder agierten als Beobachterinnen und Beobachter und gesellten sich zu einem Duo hinzu. (Hyun & Yoon, 2009, S. 678 f.)

4.4.2 Exemplarische Bodenroboter

Der Roamer als Bodenroboter repräsentiert eine Schildkröte, als Weiterentwicklung der ursprünglichen LOGO-Hardware. Papert selbst empfiehlt den Roamer als bestes didaktisches Hilfsmittel zur Vermittlung der Programmierung. (Csink & Farkas, 2010, S. 440)

Csink & Farkas (2010) klassifizieren den Roamer als ideal für einen Einstieg in die Programmierung und speziell für junge Kinder als geeignet. (Csink & Farkas, 2010, S. 440) Verpackt in ein Spiel, bei dem der Roboter von den Kindern programmiert und gesteuert werden muss, können auch junge Kinder aktiv erste Programmiererfahrungen sammeln. (Csink & Farkas, 2010, S. 441)

Der Curlybot, entwickelt für Kinder ab 4 Jahren, ist ein autonomes Fahrzeug mit zwei Rädern, welches über eine integrierte Elektronik in der Lage ist zu erkennen, in welche Richtung es bewegt wird. Die Kinder können mit dem Curlybot die ersten Schritte in der Programmierung durch „Programming by Example“ vornehmen. Dazu steht ihnen am Curlybot eine kleine Taste zur Verfügung, die entweder zum Aufnehmen oder zum Abspielen einer beliebigen von den Kindern gewählten Bewegungsreihenfolge dient. (Frei, Su, Mikhak & Ishii, 2000, S. 129 ff.) Um mit dem Curlybot geometrische Figuren zu zeichnen, kann auch ein Stift montiert werden. (Frei et al., 2000, S. 130)

Erste Evaluationen von Frei et al. (2000) zeigten, dass Kinder sehr schnell lernen, mit dem Curlybot umzugehen und ihn richtig zu bedienen. Meistens geschieht dies durch Zuschauen bei anderen Kindern und anschließendem Ausprobieren. Ein Viertel der beobachteten Kinder zeichnete von sich aus geometrische Figuren. Die übrigen Kinder fuhren frei gewählte Strecken, die nicht einer der allgemein bekannten geometrischen Figuren zugeordnet werden konnten. (Frei et al., 2000, S. 133) Bei der Evaluierung fiel zugleich auf, dass die meisten Kinder den Curlybot viel zu schnell und über viel zu große Strecken bewegten. Das führte zu Irritationen bei den Kindern, da die verbaute Technik nicht in der Lage war sehr schnelle und weiträumige Bewegungen zu erfassen und in einer ähnlichen Dynamik wie bei der Aufzeichnung wiederzugeben. Die Kinder hatten dennoch viel Spaß bei der Nutzung des Curlybot. (Frei et al., 2000, S. 134)

¹⁹ Anmerkung: Ein iRobiQ kann von Kindern nicht programmiert werden.

Die populären Lego® Mindstorms® Roboter existieren historisch betrachtet in drei Versionen: Lego® Mindstorms® RCX (1998, Version 1), Lego® Mindstorms® NXT (2006, Version 2) sowie Lego® Mindstorms® EV3 (2013, Version 3).²⁰

Mit NQCBaby ist es möglich durch Einsatz von LOGO als Programmiersprache einen einfachen RCX-Roboter zu steuern. Im Vordergrund steht dabei die Möglichkeit für jüngere Kinder mittels natürlicher Sprache, angelehnt an LOGO, einen Roboter zu steuern. (Demo & Marcianó, 2007, S. 46), (Demo, Marcianó & Siega, 2008) Die nächste Software-Version von NQCBaby unterstützt NXT-Roboter und so eine noch größere Anzahl an Robotern. (Demo & Marcianó, 2007, S. 55)

Garcia-Sanjuan, Jaen, Nacher & Catala (2015) entwickelten auf Basis eines Lego® Mindstorms® EV3 einen eigenen Bodenroboter für Kindergartenkinder der mittels RFID Tags auf selbstgebastelten Schilderchen gesteuert werden kann. (Garcia-Sanjuan et al., 2015, S. 3 f.)

Für einen vielfältigen Einsatz in der Schule wurden die LEGO® WeDO Roboter nicht nur zur Vermittlung von Programmierung und Robotik entwickelt, sondern auch als interdisziplinäres Lehr-Lernmaterial²¹ für die Themen Energie, Mechanik, Naturwissenschaften, aber auch Mathematik und Sprachen. (Ma, Prejean & Williams, 2011, S. 3699) Die Programmierung erfolgt mittels grafischer Bedienoberfläche am Computer und wird mittels USB an den Roboter übertragen. (Mayerová, 2012, S. 33 f.) In der Version 2 verfügt der Roboter zusätzlich über eine Bluetooth-Schnittstelle und kann somit auch mit gängigen Tablets und Smartphones via App gesteuert und programmiert werden.²²

LEGO® WeDO Roboter werden entweder mit Scratch programmiert, was sich eher für Kinder ab 9 Jahren eignet (Mayerová, 2012, S. 34) oder z.B. mit CHERP, das für 4- bis 7-jährige Kinder entwickelt wurde (siehe auch Kapitel 4.3.2.3). (Strawhacker & Bers, 2014, S. 298), (Sullivan, Kazakoff & Bers, 2013, S. 209)

Für junge Kinder gut geeignet sind Bee-Bots und ProBots. Das sind Bodenroboter, die kommerziell vertrieben werden und jungen Kindern den Einstieg in die Programmierung ermöglichen sollen. Der Bee-Bot, ein Roboter in Bienen-Form, kann durch Druck auf die Tasten, die am Rücken des Roboters zur Verfügung stehen, programmiert werden. Bis zu 40 Befehle können gespeichert werden, die sich auf vorwärts, rückwärts, links drehen (90 Grad), rechts drehen (90 Grad) und Pause beschränken. Der Pro-Bot stellt, insbesondere durch die Verwendung von Sensoren, mehr Funktionalitäten zur Verfügung. (Sullivan, Kazakoff & Bers, 2013, S. 205) Beide Roboter können ohne Computer direkt am Roboter programmiert werden. Der Constructa-Bot unterscheidet sich technisch nicht vom Bee-Bot, lediglich seine äußere Form erinnert nicht an eine Biene sondern an ein Baufahrzeug. Der Blue-Bot²³ erweitert den Bee-Bot um eine Bluetooth-Schnittstelle, so dass der Blue-Bot auch über Apps am Smartphone, Tablet oder PC programmiert werden kann. Die App bietet als zusätzliche Funktionalitäten 45 Grad Drehungen und eine Wiederholung von Anweisungen an, die am Blue-Bot selbst über die Hardware Benutzerschnittstelle nicht eingegeben werden können.

Pekárová (2008) stellt Vorschulkindern (im Alter von 5 bis 6 Jahren) inspiriert von Curricula in Großbritannien und Australien, wo die Einbindung von Roboter-Spielzeug zur Unterstützung des Lernens bereits erfolgt ist und zum schulischen Alltag gehört, Bee-Bots zur Verfügung. „The curriculum introduces programmable toys as a good example for developing knowledge and understanding of the contemporary world.“ (Pekárová, 2008, S.

²⁰ <http://www.lego.com/en-us/mindstorms/history>, 23.04.2018

²¹ <https://education.lego.com/de-de/lesi/elementary/wedo-2/software>, 23.04.2018

²² <https://education.LEGO.com/de-de/lesi/elementary/wedo-2/software-info>, 09.06.2018

²³ <https://www.terrapinlogo.com/robots.html>, 08.06.2018

113) Die Bee-Bots verfügen über mehrere Auszeichnungen, die ihren Einsatz in Kindergärten und Vorschulen rechtfertigen. (Pekárová, 2008, S. 114) Darüber hinaus basiert die Programmierung der Bee-Bots auf LOGO, das sich im Einsatz mit Vorschulkindern und jungen Schulkindern bereits bewährt hat. (Pekárová, 2008, S. 114) Auch Beraza, Pina & Demo (2010) nutzen Bee-Bots mit jungen Kindern, wie viele andere auch. Das bestätigt die hohe Polarität des Bee-Bots im Praxiseinsatz mit jungen Kindern.

5 Child-Centered-Design

„Das heute selbstverständliche Konzept der graphischen Benutzeroberfläche, welches die Bedienung von Computern wesentlich vereinfacht hat, ist ein bekanntes Beispiel für Einfachheit dank begrifflicher Klarheit.“ (Nievergelt, 1999, S. 364)

Neben einer Einführung in das Thema Usability für Kinder betrachtet die Verfasserin der gegenständlichen Arbeit zu Beginn die charakteristischen Merkmale und Eigenschaften der Zielgruppe Kinder und legt gesammelte Gestaltungsrichtlinien sowie Guidelines für diese Zielgruppe dar. (Kapitel 5.1)

Die Zusammenarbeit mit Kindern zur Gestaltung von altersadäquaten Produkten kann auf unterschiedliche Art und Weise erfolgen und auch in unterschiedlicher Intensität. Dazu stellt die Autorin bekannte und zum Teil auch weniger bekannte Ansätze zur Integration von Kindern im Alter von 3 bis 6 Jahren in das Usability Engineering vor. (Kapitel 5.2)

Zur Evaluation der Usability stehen bewährte Methoden zur Verfügung, die jedoch nicht alle gleich gut für junge Kinder geeignet sind. Die spezifischen Forschungsergebnisse für die Zielgruppe der 3- bis 6-jährigen Kinder im Einsatz bewährter Methoden stellt die Autorin in Kapitel 5.3 vor.

5.1 Einführung

Gute Usability ist heute von hoher Bedeutung für den Erfolg von Produkten. Gerade bei Software für Kinder, die zum Lernen und Erlernen von neuen Inhalten dient, muss auf gute Usability geachtet werden, damit Lernende nicht von dem zu lernenden Inhalt abgelenkt werden und gleichzeitig der Lernprozess positiv beeinflusst wird. (Leung, 2006, S. 543), (MacFarlane, Sim & Horton, 2005, S. 108) Ebenso bei Programmierumgebungen spielt die Usability eine große Rolle, um effizient neue Programme entwickeln zu können. (Anslow, Markstrum & Murphy-Hill, 2009, S. 1053)

Um gute Usability zu erreichen, sind nicht nur bekannte Standards und Richtlinien zu berücksichtigen, sondern auch die entsprechenden Testmethoden einzusetzen. (Reed, Holdaway, Isensee, Buie, Fox, Williams & Lund, 1999, S. 128 f.) Die Art und Weise der Interaktionen mit grafischen Benutzeroberflächen, die Nielsen schon 1993 unter dem Begriff der „Satisfaction“ zusammenfasste (Nielsen, 1993, S. 26 f.), spielen eine große Rolle für den Erfolg selbiger (Fernaesus & Tholander, 2006, S. 1014). Neue Herausforderungen brachte die Kategorie der mobilen Geräte mit sich, die mit genauen Zielgruppenanalysen und adaptiven Interaction- sowie Interface-Designs gemeistert werden können. (Su Kuen Seong, 2006, S. 25)

Bei der Entwicklung einer zielgruppengerechten Benutzerschnittstelle ist vor allem die Sichtweise der zukünftigen Benutzerinnen und Benutzer zu berücksichtigen, um Missinterpretationen des Systemverhaltens zu vermeiden. (Nievergelt, 1999, S. 364) Die Unterschiede in der Sichtweise sind umso größer, wenn es sich bei der Entwicklerin, dem Entwickler um eine Erwachsene, einen Erwachsenen und bei der Benutzerin, dem Benutzer um ein Kind handelt. Hier gilt es besonders die metakognitiven Vorstellungen der Entwicklerinnen und Entwickler mit den metakognitiven Vorstellungen der Benutzerinnen und Benutzer in Einklang zu bringen. (Blackwell, 1996, S. 240) Die große Herausforderung bei der Gestaltung von visuellen Programmierumgebungen ist die Vielzahl an Icons, die gerade bei visuellen Programmierumgebungen, die ohne Sprache auskommen sollen, benötigt wird. (Repenning, 1994, S. 296 ff.)

„When designing for children it is important to keep in mind that they perceive and make sense of the world in a significantly different way than adults. [...] An adult typically sees a book as something to be read, whereas a child might see it as something to build with or hide behind.“ (Jensen, Bursleson & Sadauskas, 2012, S. 52)

5.1.1 Die Zielgruppe Kinder

Kinder sind eine spezielle Zielgruppe, die sich von Erwachsenen deutlich in ihrer Motivation, ihren Wünschen und Erwartungen unterscheidet. (Read, MacFarlane & Casey, 2006, S. 81)

„Die großen Unterschiede zwischen Erwachsenen und Kindern im Bereich der kognitiven, motorischen, sozialen und emotionalen Fähigkeiten“ lassen die intensive Auseinandersetzung mit der Benutzerfreundlichkeit interaktiver Produkte für Kinder, neben der sehr wichtigen Berücksichtigung des pädagogischen Hintergrunds, als sehr bedeutend erscheinen. (Liebal & Exner, 2011, S. vii)

Die Integration der Kinder in den Gestaltungsprozess ist ein wesentlicher Erfolgsfaktor für das neu zu entwickelnde Produkt. Wie Nettet & Large (2004) aus der Literatur zusammenfassen, können Kinder Ideen in den Gestaltungsprozess einbringen, auf die Erwachsene vielleicht nicht gekommen wären. Genauso wie multidisziplinäre Teams immer häufiger werden, ist auch das Mitarbeiten von Kindern in Design-Teams, die Produkte für Kinder gestalten, immer wichtiger. Durch die Kinder, die die späteren Benutzerinnen und Benutzer darstellen, kommen wichtige Informationen direkt ins Design-Team. (Nettet & Large, 2004, S. 150)

Auch die Einbeziehung von Lehrkräften, z.B. in Schulen, in Evaluationssettings von Produkten, kann erkenntnisreich sein, da Lehrkräfte die Aktionen und Aussagen von Kindern sehr gut deuten und so die Evaluation gut unterstützen können. Darüber hinaus können die Lehrkräfte optimal als Vermittlerinnen und Vermittler zwischen den Kindern und den Designerinnen und Designern agieren. (Pardo, Vetere & Howard, 2006, S. 92)

Im Kontext der Evaluierung und des Tests von Produkten auf ihre Usability teilen Markopoulos, Read, MacFarlane & Höysniemi (2008) Kinder, nach Piagets Stufen der kognitiven Entwicklung, in zwei Alterskategorien ein: junge Kinder und ältere Kinder. Kinder bis 6 oder maximal 7 Jahren fallen hierbei in die Kategorie „junge Kinder“ und Kinder ab ca. 8 Jahren werden in der Gruppe „ältere Kinder“ zusammengefasst. (Markopoulos et al., 2008, S. 4) Die Unterteilung der Kinder in Altersgruppen alleine ist noch nicht ausreichend, da die Entwicklung der Kinder sehr kultur- und gesellschaftsabhängig ist, sie dient aber als erster Ansatzpunkt. (Markopoulos et al., 2008, S. 7)

„You may find that children of the same age from two different schools will behave quite differently in your evaluation studies.“ (Markopoulos et al., 2008, S. 7)

5.1.1.1 Das PLU-Modell

Das von Markopoulos et al. (2008) entwickelte PLU-Modell zeigt, wie Kinder mit Technologien interagieren. Kinder nehmen in der Interaktion mit Technologie drei verschiedene Rollen ein.

1. Kinder als Spielende (Player)
Kinder nehmen die interaktiven Produkte in dieser Rolle als Spielzeug wahr. Um Kinder zu begeistern, muss das Produkt amüsieren oder unterhalten.
2. Kinder als Lernende (Learner)
Interaktive Produkte werden in dieser Rolle als Ersatz für Lehrerin und Lehrer oder die Schule gesehen. Die Produkte müssen anleiten, herausfordern und belohnen.
3. Kinder als Benutzerin und Benutzer (User)
In dieser Rolle sehen Kinder die interaktiven Produkte als Werkzeuge. Hierzu müssen die Produkte den Kindern die Möglichkeit geben, Aufgaben einfach zu lösen.

(Markopoulos et al., 2008, S. 28)

Die Korrelation aus den Anforderungen an interaktive Produkte und die daraus resultierenden Funktionalitäten ist aus dem grafischen PLU-Modell abzuleiten (siehe Abbildung 4). Die Anforderungen der drei Rollen werden als drei Dimensionen im Raum repräsentiert, so dass nun eine Kategorisierung und Zuordnung einzelner Produkte zu den

Bedürfnissen der Kinder erfolgen kann. (Markopoulos et al., 2008, S. 28 f.) In dem abgebildeten Beispiel symbolisiert B ein Produkt, das ein Kind einsetzen möchte. Die abgebildeten Produkte C und A weisen jedoch andere Charakteristika auf, als ein Produkt B, das den Anforderungen des Kindes entsprechen würde: A ist ein eher unterstützendes Produkt und C ein Produkt, das eher Lernen mit einem geringen Spielanteil verknüpft. Je größer der Unterschied (grafisch hier als Abstand auszumachen) zwischen dem intendierten Einsatzzweck des Kindes und einem betrachteten Produkt ist, desto größer ist die Wahrscheinlichkeit, dass dieses Produkt nicht die Bedürfnisse und die Erwartungshaltung des Kindes erfüllt.

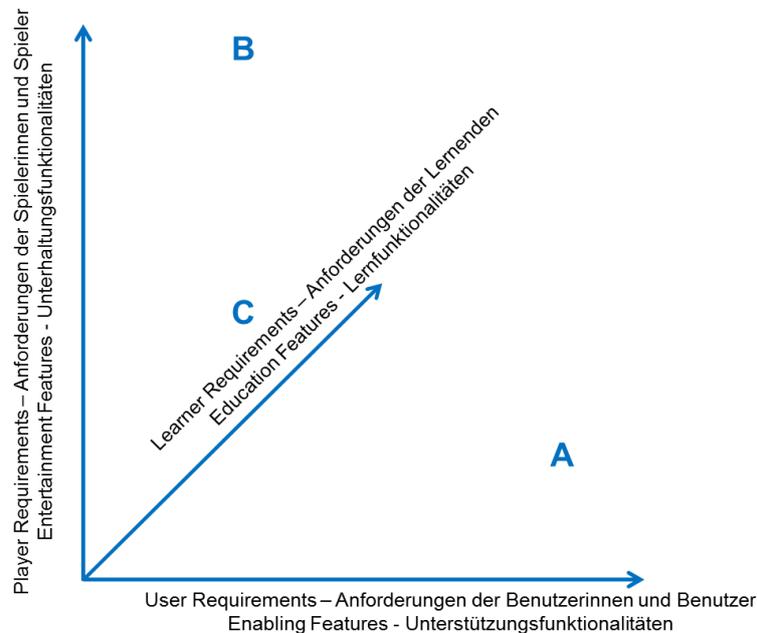


Abbildung 4: Das PLU-Modell, eigene Darstellung nach (Markopoulos et al., 2008, S. 29)

5.1.1.2 Kognitive Fähigkeiten

Auch wenn in den letzten Jahren viele Nachuntersuchungen zu Piagets Theorien durchgeführt wurden (Berk, 2005, S. 299 ff.), und einige von ihm gefundene „Mängel“ im Denken von jungen Kindern abgeschwächt und mit Einschränkungen versehen wurden, so geben diese „Mängel“ im Kontext der gegenständlichen Arbeit doch Anhaltspunkte für das Verständnis mancher Annahmen und Herangehensweisen junger Kinder im Umgang mit Hard- und Software.

„Das Unvermögen, die symbolischen Ansichten anderer von den eigenen zu unterscheiden“ (Berk, 2005, S. 297) wird als Egozentrismus definiert. Kinder gehen davon aus, dass andere die Welt genauso wahrnehmen wie sie selbst und in Folge dessen genauso denken und fühlen. (Berk, 2005, S. 297) In diesem Zusammenhang wird in der Literatur oftmals auf das „Drei Berge Problem“ verwiesen, welches die Unzulänglichkeiten von jungen Kindern im Einnehmen einer anderen Perspektive darlegt: Die Kinder sehen drei Berge vom Standort A und werden aufgefordert zu schildern, wie die drei Berge aus einem anderen Standort B aussehen würden. Hierbei beschreiben junge Kinder oftmals die eigene Perspektive von Standort A anstelle des Standorts B, da sie davon ausgehen, dass alle die Welt so wahrnehmen wie sie selbst. (Mazuka, Jincho & Oishi, 2009, S. 68)

Korrelierend mit dem Egozentrismus ist für Piaget das Animistische Denken (Magische Denken) von Kindern in der präoperationalen Phase (siehe Kapitel 2.1.3) zu sehen. Dieses Denken „beinhaltet die Annahme, dass unbelebte Dinge lebensähnliche Merkmale haben wie Gedanken, Wünsche, Gefühle und Absichten.“ (Berk, 2005, S. 297)

Kinder in der präoperationalen Phase sind nicht in der Lage, Invarianz zu erkennen. Hierbei handelt es sich um das Phänomen, dass auch wenn sich ein Objekt äußerlich verändert, seine physikalischen Eigenschaften dennoch gleichbleiben. (Berk, 2005, S. 297) Der Begriff der Zentrierung geht mit dem Nichterkennen der Invarianz einher, da sich Kinder im Alter von 2 bis 7 Jahren auf einen Aspekt konzentrieren und dadurch andere Aspekte vernachlässigen, wenn sie eine Situation betrachten. (Berk, 2005, S. 298)

In weiterer Folge ist nach Piaget die Irreversibilität bei jungen Kindern stark ausgeprägt. Darunter ist das Unvermögen zu verstehen, eine Reihe von Schritten, die zur Lösung eines Problems führen, mental in reversibler Reihenfolge zurück zum ursprünglichen Problem auszuführen. Bei jeder logischen Operation ist Reversibilität von großer Bedeutung. (Berk, 2005, S. 298 f.) Schwierigkeiten, bei unterschiedlichen Versuchen eine Aufgabe zu lösen, die Meinung bzw. das Vorgehen zu ändern, können mit diesem Phänomen in Zusammenhang gebracht werden. (Mazuka, Jincho & Oishi, 2009, S. 74)

Der „Mangel“ an logischem Denken führt bei jungen Kindern zu Schwächen in der hierarchischen Klassifikation, wo es darum geht, Objekte aufgrund von Ähnlichkeiten und Unterschieden zu klassifizieren. (Berk, 2005, S. 299) Halford & Andrews (2006) betonen, dass nach der CCC- (cognitive complexity and control) Theorie junge Kinder in der Lage sind, einfache „Wenn-Dann“-Regeln aufzustellen. Sie können diese allerdings nicht zu einer hierarchisch übergeordneten komplexeren Regel zusammenfassen. (Halford & Andrews, 2006, S. 563 f.)

Ergänzend führen Mano & Campos (2005) noch folgende Aspekte an, die zu berücksichtigen sind: Kinder bewerten und argumentieren oftmals intuitiv aus einem externen Blickwinkel. Wenn die von ihren Sinnen wahrgenommenen Fakten nicht mit ihrem Mentalmodell übereinstimmen, dann können Sie die Situation nicht korrekt einschätzen. Außerdem haben Kinder Schwierigkeiten damit, verschiedene Aspekte oder Teile eines Objekts oder einer Situation zu trennen und sehen sie als untrennbar an. Darüber hinaus haben junge Kinder Schwierigkeiten Reihenfolgen zu ordnen oder Serien zu erstellen. (Mano & Campos, 2005, S. 1)

Um mit Software erfolgreich arbeiten zu können und auch die Wege in den vielen mit „Finde den Weg“-Aufgabenstellungen gestalteten Programmierumgebungen für junge Kinder zu finden, ist es von Nöten, dass Kinder über grundsätzliche mathematische Fähigkeiten wie zählen, Zahlenverständnis, Zahlentransformationen, schätzen und Zahlenmuster erkennen verfügen. (Jordan et al., 2006, S. 154 f.) Als herausfordernd beim Planen eines Weges sind exemplarisch die korrekte Verwendung von Zahlwörtern und das Zählen in Einerschritten zu nennen. Dies stellt oftmals ein Problem für junge Kinder dar, da hierbei viele verschiedene Fehler auftreten können, beispielweise das Nennen von zwei Zahlwörtern für ein Objekt, ein Objekt wird mehrmals gezählt, uvm. (Geary, 2007, S. 785 f.)

Gerade die Altersgruppe der jungen Kinder verfügt nach Markopoulos et al. (2008) über ein sehr hohes Selbstbewusstsein, welches eng mit der emotionalen Entwicklung des Kindes verknüpft ist. Das Selbstbewusstsein der älteren Kinder (im Speziellen der 9- bis 10-jährigen) hingegen ist deutlich unterrepräsentiert, nur mehr 15 % der Kinder dieses Alters haben ein ausgeprägtes Selbstbewusstsein. (Markopoulos et al., 2008, S. 13) Dies macht die Gruppe der jungen Kinder sehr interessant für Evaluierungen und Tests, da davon ausgegangen werden darf, dass sich diese Kinder aktiv einbringen und nicht vor möglichen Fehlern zurückschrecken.

5.1.1.3 Entwicklung der Motorik

Im Hinblick auf den Einsatz von IKT wird im Folgenden die Entwicklung der für die Bedienung von Hard- und Softwareprodukten notwendigen grob- und feinmotorischen Fähigkeiten von jungen Kindern dargelegt.

5.1.1.3.1 Grobmotorische Fähigkeiten

2- bis 3-jährige Kinder können mit steifem Oberkörper werfen und fangen, auch ist es ihnen möglich ein Kinderfahrzeug ein wenig zu steuern. Das Werfen erfolgt bei Kindern mit 3 bis 4 Jahren schon mit leichter Beteiligung des Oberkörpers, gefangen wird, indem der Ball gegen die Brust gedrückt wird, Dreiräder können gesteuert werden. Kinder im Alter von 4 bis 5 Jahren fangen einen Ball mit den Händen, und beim Werfen wird der Körper bereits gedreht, das Dreiradfahren ist auch bei hohem Tempo möglich und dieses wird geschickt gesteuert. Für 5- bis 6-Jährige können schon Fahrräder (mit Stützen) zum Einsatz kommen, Seilspringen wird erlernt und das Werf- und Fangverhalten zeigt Reife. (Berk, 2005, S. 291)

5.1.1.3.2 Feinmotorische Fähigkeiten

Kinder im Alter von 2 bis 3 Jahren sind in der Lage, einfache Kleidungsstücke an- und auszuziehen, lange Reißverschlüsse zu nutzen und mit einem Löffel zu essen. Ab einem Alter von 3 bis 4 Jahren können Kinder Dreiräder fahren, große Knöpfe öffnen und schließen, alleine essen, eine Schere verwenden, einfache Bilder von Menschen malen und vertikale Linien und Kreise kopieren. Das Benützen von einer Gabel, Linien genau mit einer Schere schneiden und Kopieren von ersten Buchstaben, eines Dreiecks und Kreuzes ist Kindern ab 4 bis 5 Jahren möglich. 5- bis 6-Jährige verwenden erfolgreich Messer um weiche Nahrungsmittel zu schneiden, können ein Schuhband eigenständig binden, kopieren einige Zahlen und einfache Wörter und die Zeichnungen von Menschen sind bereits etwas komplexer. (Berk, 2005, S. 291)

Angesichts dieser umfangreichen Fähigkeiten über die Kinder bereits im Kindergartenalter verfügen, kann unter Berücksichtigung der Physionomie bei der Auswahl der Produkte abgeleitet werden, dass Kindergartenkinder unter Anleitung von ihrer Motorik her in der Lage sind, Hard- und Softwareprodukte adäquat zu bedienen. Diese Bedienung unterliegt natürlich ebenso wie etwa das Erlernen des Fahrradfahrens, oder das Benützen einer Schere einem gewissen Lernprozess.

5.1.2 Guidelines und Rahmenbedingungen

Nachfolgend finden sich zahlreiche aus der Literatur konsolidierte Gestaltungsrichtlinien zur Entwicklung von Benutzerschnittstellen für Kinder, mit besonderem Augenmerk auf Programmierumgebungen.

5.1.2.1 Grundlagen zur Zielgruppe

Es herrscht im Forschungsbetrieb Einigkeit darüber, dass die zukünftigen Benutzerinnen und Benutzer von Anbeginn an im Fokus des Interesses stehen sollen. Sie spielen eine wichtige Rolle in der Analysephase und sollen vor anderen Aspekten bedacht werden um ihre Bedürfnisse und Anforderungen in der Entwicklung berücksichtigen zu können. (Su Kuen Seong, 2006, S. 27)

Gelderblom & Kotzé (2008) stellen auf Basis der Theorien der kognitiven Entwicklung des Kindes 35 Richtlinien zur Gestaltung von Technologie für Kinder vor. Dazu haben sie diese in sechs Kategorien zusammengefasst:

1. Richtlinien zur Entwicklung von allgemeinen Fähigkeiten
2. Richtlinien zur Entwicklung von spezifischen Fähigkeiten
3. Richtlinien zur altersadäquaten Gestaltung
4. Richtlinien zur Unterstützung der Benutzerinnen und Benutzer
5. Richtlinien zu Kultur, Kontext und Vorwissen
6. Richtlinien zur Unterstützung der (anhaltenden) Beschäftigung

(Gelderblom & Kotzé, 2008, S. 72)

Darüber hinaus haben Gelderblom & Kotzé (2009) 350 weitere Richtlinien für die Entwicklung von Technologie für Kinder entwickelt und präsentiert. Die wichtigsten Regeln haben sie in zehn sog. Lektionen zusammengefasst:

1. Wenn ein Kind ein spezifisches Problem in einer Domäne lösen kann, dann kann es nicht zwingendermaßen diese Fähigkeiten auf eine andere Domäne übertragen. (Gelderblom & Kotzé, 2009, S. 54)
2. Junge Kinder finden es schwierig zwischen dem formalen System der Mathematik und den Quantitäten, Operationen und Konzepten, die sie repräsentieren, zu übersetzen. (Gelderblom & Kotzé, 2009, S. 54 f.)
3. Trenne nicht den erklärenden Teil vom Spaß-Teil eines Produkts, sonst glauben Kinder, dass der Spaß erst nach der Erklärung kommt und die Erklärung selbst keinen Spaß macht. (Gelderblom & Kotzé, 2009, S. 55)
4. Erlaube Kindern verschiedene Strategien zur Problemlösung. (Gelderblom & Kotzé, 2009, S. 55 f.)
5. Fördere das Reflektieren und die Entwicklung von Fähigkeiten von Kindern um ihnen die zu Grunde liegenden Prozesse für Erfolg oder Misserfolg bewusst zu machen. (Gelderblom & Kotzé, 2009, S. 56)
6. Eine Fähigkeit kann bei Kindern unterschiedlicher kultureller Gruppen auf unterschiedliche Art und Weise gelehrt oder erworben werden. (Gelderblom & Kotzé, 2009, S. 56 f.)
7. Verlasse Dich nicht darauf, dass Kinder Anweisungen, die sie gehört haben, genau wiedergeben können. (Gelderblom & Kotzé, 2009, S. 57)
8. Kindern ist die Kontrolle darüber zu geben, wie oft und in welchem Ausmaß sie verbales Feedback erhalten möchten. (Gelderblom & Kotzé, 2009, S. 57)
9. Junge Kinder können Point & Click Aktionen schneller und präziser ausführen, als Drag & Drop Aktionen. (Gelderblom & Kotzé, 2009, S. 57 f.)
10. Die Erreichbarkeit von favorisierten Features, Vertrautheit und Multi-Modalität haben eine spezielle Bedeutung für Produkte für Kinder. (Gelderblom & Kotzé, 2009, S. 58)

Im Kontext von (Lern-)Software für Kinder, die sich gesundheitlich bedingt länger im Krankenhaus aufhalten, haben Hoiseth, Giannakos & Jaccheri (2013) weitere Richtlinien entwickelt, die im Folgenden verallgemeinert wurden und generell für die Gestaltung von Produkten für Kinder herangezogen werden können:

1. Für die jeweilige Situation sind die entsprechenden Spielaktivitäten anzubieten, um relevante Informationen rund um den Lerngegenstand im jeweiligen Kontext zu vermitteln. (Hoiseth, Giannakos & Jaccheri, 2013, S. 454)
2. Familiäre und soziale Aktivitäten sollen unterstützt werden. (Hoiseth, Giannakos & Jaccheri, 2013, S. 454)
3. Es sind Geschichten zu wählen, die ein Zusammengehörigkeitsgefühl vermitteln. (Hoiseth, Giannakos & Jaccheri, 2013, S. 454)
4. Verwende wiederholende Elemente, um Routine und Wiedererkennbarkeit zu fördern. (Hoiseth, Giannakos & Jaccheri, 2013, S. 454)
5. Setze Elemente ein, die als bedeutungsvolle Anreize dienen können, um die Motivation hoch zu halten. (Hoiseth, Giannakos & Jaccheri, 2013, S. 454 f.)
6. Vermische in angemessener Weise Realität und Fantasie, im Speziellen bei der Auswahl der Charaktere. (Hoiseth, Giannakos & Jaccheri, 2013, S. 455)
7. Stelle praktische und informative Informationen zum Lerngegenstand zur Verfügung. (Hoiseth, Giannakos & Jaccheri, 2013, S. 455)

Liebal & Exner (2011) haben 110 Gestaltungsempfehlungen für Benutzerschnittstellen ausgearbeitet, die, unter Berücksichtigung der Zielgruppe Kinder, unterstützend in der Software Entwicklung eingesetzt werden können. Sie haben die Gestaltungsempfehlungen in

die drei Kategorien Screendesign, Steuerung und Interaktion sowie Inhalt, aufgeteilt auf 18 Untergruppen, zusammengefasst. (Liebal & Exner, 2011, S. 124 ff.)

Screendesign	Steuerung und Interaktion	Inhalt
1. Farbe 2. Schrift und Textgestaltung 3. Bildschirmaufteilung 4. Visuelle Gestaltungselemente <ul style="list-style-type: none"> a. Bilder und Grafiken b. Animationen c. Videos 5. Auditive Gestaltungselemente <ul style="list-style-type: none"> a. Ton b. Sprache c. Geräusche d. Hintergrundmusik 	6. Eingabegeräte <ul style="list-style-type: none"> a. Maus b. Tastatur c. Touchscreen 7. Interaktionstechniken <ul style="list-style-type: none"> a. Klicks und Doppelklicks b. Point & Click und Drag & Drop c. Scrollen 8. Navigation und Menü <ul style="list-style-type: none"> a. Informationsarchitektur b. Interface Metaphern c. Icons d. Schaltflächen e. Links 9. Benutzerunterstützung <ul style="list-style-type: none"> a. Feedback b. Statusanzeige c. Cursor d. Hilfe e. Instruktionen f. Suche 	10. Entertainment 11. Interaktionselemente 12. Leitfiguren 13. Text 14. Überschriften 15. Datenabfrage 16. Anmeldebereiche 17. Elternspezifische Inhalte 18. Werbung

Tabelle 1: Überblick Gestaltungsempfehlungen nach (Liebal & Exner, 2011, S. 125 - 136), eigene Darstellung

Die Ausformulierung der Gestaltungsempfehlungen für Benutzerschnittstellen für Kinder sind bei Liebal & Exner (2011, S. 136 - 190) nachzulesen.

5.1.2.2 Evaluation und Test

Um Kinder effektiv und ergebnisbringend in ein Testszenario einbinden zu können, haben Markopoulos et al. (2008) einen Fragenkatalog entwickelt, den Testleiterinnen und -leiter bei der Konzeptionierung eines Tests als Leitfaden zur Hilfe nehmen können. Dieser Fragenkatalog deckt die Themenbereiche der physischen Entwicklung des Kindes ebenso ab, wie die der sozioemotionalen oder der kognitiven. (Markopoulos et al., 2008, S. 11 ff.) Die frei übersetzte Liste des Leitfadens ist Tabelle 25 in Anhang A1.1 zu entnehmen.

Bei Hanna, Ridsen & Alexander (1997) finden sich weitere Vorschläge und Empfehlungen zur Gestaltung von Usability-Tests und Evaluationsszenarien, wobei der Schwerpunkt auf der Organisation von Usability-Tests liegt. Eine von der Verfasserin der gegenständlichen Arbeit gekürzte und zusammengefasste Darstellung der Handlungsempfehlungen findet sich in Tabelle 26 in Anhang A1.2.

Zur Untersuchung der Bedienbarkeit und Nutzbarkeit von Software bieten Standards und Normen eine gute Grundlage für die Bewertung selbiger. Bei Übereinstimmung in allen (funktionalen) Aspekten ist die Software leichter zu bedienen, die mit Usability-Richtlinien konform geht. (Reed et al., 1999, S. 124) Eine geeignete Methode für eine derartige Untersuchung ist die heuristische Evaluation, siehe Kapitel 5.3.2.

Sarodnick & Brau (2011) führen die wichtigsten Normen und Standards im Kontext der Usability an. Eine der bekanntesten und umfangreichsten Sammlungen an Normen ist die DIN EN ISO 9241 zur Ergonomie der Mensch-System-Interaktion. Bei der ISO/IEC TR 25060 handelt es sich um ein Rahmenwerk für Normen hinsichtlich der Ergebnisse des Usability Engineering-Prozesses. Die ISO/IEC 25062 stellt eine standardisierte Methode zur

Verfügung, um über die Ergebnisse summativer Usability-Evaluationen zu berichten. Die ISO 9355 beschäftigt sich mit den ergonomischen Anforderungen und die DIN EN ISO 14915 mit der Software-Ergonomie für Multimedia-Benutzungsschnittstellen. Zahlreiche weitere Normen, auch auf nationaler und nicht nur internationaler Ebene, können zur Evaluierung und Arbeit im Bereich Usability herangezogen werden. (Sarodnick & Brau, 2011, S. 34 - 48)

Um sich nicht nur auf Standards zu verlassen, können sich Designerinnen und Designer in Abhängigkeit des Fortschritts der jeweiligen Produktentwicklung Evaluationsfragen stellen (Markopoulos et al., 2008, S. 72), die in Abbildung 39 in Anhang A1.3 dargelegt sind.

5.1.2.3 Interface – Grafische / Textbasierte / Haptische Benutzerschnittstelle

Kindern muss es ermöglicht werden, den direkten Zusammenhang zwischen den eingesetzten (visualisierten) Befehlen und deren zugrunde liegenden Konzepten verfolgen und erkennen zu können. So kann das Verständnis der Funktionalität von Befehlen und Befehlsabfolgen gefördert werden. Durch Hilfestellungen soll nicht korrekten Befehlsabfolgen entgegengewirkt werden. (Sheehan, 2003, S. 81 f.)

Das Interface der Programmierumgebung soll bei der Verwendung von Farben, Icons und weiteren gestalterischen Elementen, sowie bei der Platzierung von Buttons etc. durch alle Anwendungsfälle hindurch konsistent sein. (Su Kuen Seong, 2006, S. 28)

Basierend auf Piagets Theorie der Entwicklung von jungen Kindern (siehe Kapitel 2.1.3 und Kapitel 5.1.1.2), haben Mano & Campos (2005) zehn Fragen, die sich Designerinnen und Designer bei der Entwicklung von Benutzerschnittstellen für junge Kinder stellen sollten, erarbeitet:

1. Ist es Kindern möglich, einen Bild-Link zu erkennen oder ist eine Kombination aus Bild und Text für einen Link effizienter?
2. Ist es Kindern möglich eine Tastatur innerhalb der Applikation zu nutzen?
3. Muss die Benutzerschnittstelle willkürlich bedienbar sein (um kein falsches Lernen oder einfaches Auswendiglernen von Ursache-Wirkungs-Beziehungen hervorzurufen) oder das Erlernen einer Sequenz von Ereignissen erlauben?
4. Sind Kinder in der Lage die korrekte Verbindung zwischen Bildern und den auftretenden Aktionen herzustellen? (In Verbindung mit Frage 1)
5. Ist es Kindern möglich zu erkennen, wann eine Aktion widerrufen werden muss?
6. Sind Kinder in der Lage erfolgreiche Aktionen innerhalb der Benutzerschnittstelle nochmals auszuführen?
7. Sind Kinder in der Lage Aufgaben zu lösen, die mehrere unabhängige Aktionen erfordern?
8. Sind Kinder in der Lage die Hilfe aufzurufen oder muss die Hilfe bereits durch die Benutzerschnittstelle angezeigt werden?
9. Muss die Anzahl an interaktiven Kontrollelementen auf ein Minimum reduziert sein?
10. Kann ein Kind eine Benutzerschnittstelle verstehen, die in Kategorien unterteilt ist?

(Mano & Campos, 2005, S. 2)

Beim Einsatz von Tangible User-Interfaces sind noch weitere Aspekte zu beachten. Unter anderem:

- **Physische Ergonomie:** Die haptische Benutzerschnittstelle soll an die Größe der Kinder angepasst sein und die physischen Fähigkeiten der Kinder berücksichtigen.
- **Affordanz, Aufforderungscharakter:** Stelle klare Hinweise über die Nutzung des Systems und die Manipulation von Objekten zur Verfügung. Den verwendeten Objekten und Artefakten sollte eine gute und klare Aufforderung zur Interaktion innewohnen.
- **Verspieltheit:** Beim Lernen oder Spielen haben Kinder das Bedürfnis herauszufinden, was passiert, wenn sie mit der Benutzerschnittstelle und den Objekten selbiger spielen.

(Xu, Read, Mazzone, MacFarlane & Brown, 2007, S. 1014)

5.1.2.4 Icons

Bei der Gestaltung grafischer Benutzeroberflächen, die mit zusätzlichen Zeigergeräten wie z.B. einer Maus bedient werden sollen, ist darauf zu achten, dass Kinder viereckige Ziele am Computerbildschirm leichter anvisieren können, als runde, da die Oberfläche der Ziele größer ist. Darüber hinaus gilt, je größer die Ziele sind, desto leichter und akkurater können sie von Kindern anvisiert werden. (Donker & Reitsma, 2007b, S. 603) Während es für viele Standard-Aufgaben bereits bewährte Icons gibt, sind gerade für spezielle Aufgaben und hierbei besonders für junge Kinder ohne Lesefähigkeiten Icons oftmals schwierig zu entwickeln. (McKnight & Read, 2009, S. 258) Während konkrete Objekte sehr gut für eine Darstellung als Icon geeignet sind, sind Funktionen schwer abzubilden. Es gilt möglichst konkrete Objekte zu finden, die mit den Funktionen assoziiert werden und die Funktion am ehesten repräsentieren. Dies ist allerdings nicht immer einfach und kann auch zu Missverständnissen führen. (McKnight & Read, 2009, S. 259), (Yatim, Shariff, Fathil, Zaki, Ibharm, Abdullah & Supal, 2011, S. 1)

5.1.2.5 Sprache

Bei der Entwicklung einer Programmiersprache gilt es in zweierlei Hinsicht dem HCI-Prinzip, dass Systeme die (natürliche) Sprache der Zielgruppe, hier Kinder, sprechen sollen, zu folgen. (Pane, Myers & Miller, 2002, S. 200)

Sowohl die Programmiersprache an sich (Pane, Myers & Miller, 2002, S. 200), als auch die Benutzerschnittstelle der Programmierumgebung sollen in den Bezeichnungen, Dialogen und Konzepten ähnlich zur (natürlichen) Sprache der Kinder sein. (Su Kuen Seong, 2006, S. 27)

Information soll immer dann sichtbar und leicht verfügbar im Programmiersystem zur Verfügung stehen, wenn sie benötigt wird. (Pane, Myers & Miller, 2002, S. 201)

5.1.2.6 Interaktion

Bei der Gestaltung der Benutzerschnittstelle gilt es, für die Entwicklung des Grunddesigns bewährte Standards als Grundlage anzuwenden, um für wichtige oder schwierige Interaktionspfad-Entscheidungen mehr Zeit aufbringen zu können. (Reed et al., 1999, S. 124) Bei der Zielgruppe Kinder sind besonders einfache und intuitive Interaktionspfade zu gestalten. Das „look and feel“ der Software soll vertraut und innerhalb der Software standardisiert sein. Dadurch wird es den Kindern ermöglicht, innerhalb der Software erworbene Fertigkeiten bei anderen Anwendungsfällen in der Software einzusetzen. (Reed et al., 1999, S. 124) Die Bedienbarkeit von Software soll sich an den diesbezüglichen Konzepten des zugrundeliegenden Betriebssystems orientieren. (Reed et al., 1999, S. 124), (Su Kuen Seong, 2006, S. 28)

Die Benutzerschnittstellen sollen möglichst robust gestaltet sein und Kinder sollen durch eine durchdachte Benutzerführung daran gehindert werden Fehler zu machen. Sollten dennoch

Fehler und unerwartete Ereignisse eintreten, so sind klare und verständliche Fehlermeldungen auszugeben, die, je nach Art und Gegebenheit des Fehlers, den Kindern und etwaigen Betreuungspersonen auch Lösungsstrategien zur Verfügung stellen. (Su Kuen Seong, 2006, S. 27), (Sim, MacFarlane & Read, 2006, S. 237)

Den Kindern soll stets der aktuelle Status ihrer Interaktionen mit der Software bekanntgegeben werden. Neben der zeitnahen und zielgruppengerechten Rückmeldung auf einzelne Interaktionen zur Vermeidung von Frustrationserlebnissen in der Bedienung, sollen auch detaillierte Rückmeldungen zu Lösungsversuchen von Übungen die Motivation der Kinder fördern. (Su Kuen Seong, 2006, S. 27)

Um die Kinder nicht zu überfordern und die kurze Aufmerksamkeitsspanne von 8 - 15 Minuten zu berücksichtigen, sollen die Lerninhalte in kleinen und einheitlich aufgebauten Einheiten organisiert sein. (Su Kuen Seong, 2006, S. 27)

Kinder sollen bei der Arbeit mit Programmierumgebungen dort abgeholt werden, wo ihr Vorwissen ansetzt. Die ZPD „Zone of proximal development“, (siehe auch Kapitel 2.1.5) ist die Lernsituation, in der Lernende, aufgrund ihres Vorwissens und ihrer Fähigkeiten, mit höchster Wahrscheinlichkeit (Neues) lernen. (Vygotsky, 1978)

5.1.3 Evaluation von Eingabegeräten für Kinder

Zur Bedienung von Computer-Systemen gibt es eine Vielzahl an Geräten am Markt. Grundsätzlich lassen sich diese in zwei Kategorien einteilen. Das sind zum einen direkte und zum anderen indirekte Eingabegeräte. Erstere liefern eine unmittelbar korrespondierende Rückmeldung auf die Eingabe. Zu dieser Kategorie zählen u.a. Touchscreens. Bei indirekten Eingabegeräten ist die Bedienung losgelöst von der eigentlichen Aufgabe, wie z.B. bei einer klassischen Maus. Bei dieser muss die Benutzerin, der Benutzer räumliche Informationen verarbeiten, Hand und Auge koordinieren und über eine gewisse Feinmotorik verfügen, um das Gerät der Bestimmung entsprechend zu verwenden. (Wood, Willoughby, Schmidt, Porter, Specht & Gilbert, 2004, S. 263)

Wood et al. (2004) haben mehrere Eingabegeräte auf ihre Eignung für Vorschulkinder und ihre Lehrkräfte untersucht. Vorweg muss angemerkt werden, dass es das perfekte Eingabegerät, das für alle Personen und alle Anwendungsfälle gleichermaßen gut geeignet ist, nicht gibt. (Wood et al., 2004, S. 264) Untersucht wurden die Geräteklassen Maus, Touchscreen, Touchpad und Trackball²⁴. (Wood et al., 2004, S. 265)

Bei der Evaluierung wurden sowohl von den Kindern, als auch den Lehrkräften die Zeit und die Präzision zur Lösung von Aufgaben mit den unterschiedlichen Eingabegeräten gemessen. Weiter wurde die Korrelation zwischen den kognitiven und motorischen Fähigkeiten bei jeder Probandin, jedem Probanden pro Gerät ermittelt. Abschließend wurde ein Fragebogen genutzt um die persönlichen Präferenzen zu ermitteln. (Wood et al., 2004, S. 269)

Bei den Kindern differierte die Leistungsfähigkeit je Gerätekategorie sehr stark, während diese bei den Lehrkräften kaum variierte. Die Lehrkräfte bevorzugten klar und eindeutig die klassische Maus als das für sie einfachste und präziseste Eingabegerät. Das belegten auch die Testergebnisse. Danach folgte als zweitbeliebtestes Eingabegerät der Trackball. Dieses Erkenntnis ist auch in der weiten Verbreitung dieser Hardware und ihrer historischen Entwicklung begründet. (Wood et al., 2004, S. 277)

Wenn es um einfache Zuordnung von Elementen am Bildschirm ging, war der Touchscreen für Kinder das geeignetste Eingabegerät. Jedoch verdeckte der Arm der Kinder oftmals bei der Bedienung den übrigen Bildschirm, was vor allem der Größe des Touchscreens

²⁴ Als Beispiel wurde eines dieser Geräte angeführt:

<https://www.minicute.eu/News/1491920655/1491977661/Startseite#service>, 23.04.2018

geschuldet ist. Daher bevorzugten sie den Trackball und die Maus, da hier durch ihren Arm keine Inhalte verdeckt wurden. Wobei der Test den Trackball als effizientestes und bestes Eingabegerät identifizierte. Dieses Ergebnis deckt sich mit den Aussagen der Lehrkräfte, die ebenfalls den Trackball für die Kinder empfahlen. Das Touchpad schnitt schlechter ab, da es auf Grund seiner kleineren Fläche und den eher noch grobmotorischen Fingerbewegungen die Kinder oftmals dazu veranlasste, den Finger wieder neu anzusetzen und so verloren die Kinder die Bildschirmobjekte, die sie bewegten und die Bedienung wurde unnötig kompliziert und langwierig. (Wood et al., 2004, S. 278) Ähnliches beobachteten Mansor, De Angeli & De Bruijn (2008) beim Einsatz von sog. Tabletops (Tische, deren Oberfläche einen Touchscreen beinhaltet).

Auch beim Einsatz von Stiften zur Eingabe auf einem Touchscreen haben Kinder (im Volksschulalter) gegenüber einer klassischen Maus erhebliche Probleme, wenn auch nicht so umfangreiche. (Chang, 2008, S. 535) Bei Grafiktablets zeigen sich die Herausforderungen in der Positionierung des Cursors, da dieser am Grafiktablett selbst nicht sichtbar ist und bei Berührung des Stifts mit dem Grafiktablett der Cursor am Bildschirm ggf. seine Position sprunghaft verändert. Der unmittelbare Zusammenhang dieses Effekts erschließt sich den Kindern nur schwer. (Read, MacFarlane & Casey, 2002, S. 21 f.)

Einige Studien attestierten Kindern Unzulänglichkeiten bei sog. Drag & Drop Bewegungen mit der Maus und führten dies auf die Länge der zurückzulegenden Wege zurück. Es stellte sich jedoch in weiteren Untersuchungen heraus, dass nicht die Wegstrecke, die mit der Maus zu fahren ist, das Problem darstellt, sondern die Auswahl und das Loslassen der Bildelemente. (Donker & Reitsma, 2007, S. 257) Als Alternative kann auch das „Click-Move-Click“-Verfahren eingesetzt werden. Dabei wird ein Objekt durch Mausklick ausgewählt, die Maus an den Bestimmungsort der Verschiebung bewegt und anschließend durch erneuten Mausklick das Objekt wieder losgelassen. (Donker & Reitsma, 2007b, S. 604) Dies fällt Kindern wesentlich leichter und sie erzielen damit präzisere Ergebnisse, als mit dem klassischen Drag & Drop Verfahren. (Donker & Reitsma, 2007b, S. 605)

Darüber hinaus haben McKnight & Fitton (2010, S. 240) durch eine Analyse vermeintlicher Bedienfehler von Kindern mit der Maus als Eingabegerät festgestellt, dass Betriebssysteme für die Bedienung durch Erwachsene ausgerichtet und die Erfassung der Mauseaktionen entsprechend sensibel angelegt sind. Kinder hingegen können nicht immer Klicken ohne die Maus zu bewegen, oder brauchen zwischen zwei Klicks, für die Ausführung eines Doppelklicks, mehr Zeit. Betriebssysteme benötigen daher eine robustere Interpretation der Mauseaktionen von Kindern.

Eine Anpassung der Hardware an die Bedürfnisse der Kinder erscheint unumgänglich. Im Speziellen auf die kleineren Hände ist Rücksicht zu nehmen, was sich in der Auswahl einer geeigneten (Größe, Ergonomie, ...) Computer-Maus widerspiegeln sollte. (Bingchen & Yanqun, 2010, S. 113 ff.) Als Alternative zur klassischen Maus kann der Einsatz von Zeigestiften, Touch-Pens oder gar Touchscreens in Betracht gezogen werden. (Xiajian, Danli & Hongan, 2011, S. 572 ff.) Eine Benutzeroberfläche mit größeren Symbolen als üblich erleichtert Kindern die Navigation auf dieser, da sie weniger feinmotorisch mit der Maus agieren müssen. (Agudo, Sánchez & Rico, 2010, S. 493), (Chang, 2008, S. 535)

Darüber hinaus zeigt sich, dass der Einsatz von Desktop-Computern generell zu bevorzugen ist. In einer Studie von Straker, Coleman, Skoss, Maslen, Burgess-Limerick & Pollock (2008, S. 553) wurde festgestellt, dass der Einsatz von Tablet-Computern zu deutlich mehr Verspannungen in den Schultern und Nacken führt als die Nutzung eines herkömmlichen Desktop-Computers.

5.2 Usability Engineering mit (Kindergarten-)Kindern

An den Beginn dieses Abschnitts stellt die Verfasserin der gegenständlichen Arbeit eine Definition des Begriffs Usability Engineering, welche als Grundlage für die weiteren Abschnitte dieser Arbeit dienen soll:

„Das Usability-Engineering ist der methodische Weg zur Erzeugung der Eigenschaft Usability. Es ist ein Teilprozess der Entwicklung und Gestaltung technischer Systeme und ergänzt das klassische Engineering, beispielsweise Software-Engineering, um ergonomische Perspektiven. Dabei werden Ansätze, Methoden, Techniken und Aktivitäten für einen benutzerorientierten Entwicklungsprozess bereitgestellt.“ (Sarodnick & Brau, 2011, S. 23)

„Die großen Unterschiede zwischen Erwachsenen und Kindern im Bereich der kognitiven, motorischen, sozialen und emotionalen Fähigkeiten“ lassen die intensive Auseinandersetzung mit der Benutzerfreundlichkeit interaktiver Produkte für Kinder, neben dem pädagogischen Hintergrund, als sehr wichtig erscheinen. (Liebal & Exner, 2011, S. vii)

Um zielgruppenadäquate Produkte (Software und Webseiten) erstellen zu können, empfehlen Liebal & Exner (2011) einen Child-Centered-Design-Ansatz, bei dem die Zielgruppe Kinder im Mittelpunkt der Produktentwicklung steht. Sie beschäftigen sich hierbei vor allem mit der Analyse- und Konzeptphase im Software Engineering.

„Für die Gestaltung von Software und Websites für Kinder, müssen insbesondere deren Vorlieben, Abneigungen und Bedürfnisse berücksichtigt werden. [...] Viele Designer [und Designerinnen, Anm. d. Verf.] glauben, sie hätten eine sehr gute Vorstellung hinsichtlich der kindlichen Erwartungen und Wahrnehmung. Betrachtet man diese Vorstellung jedoch genauer, so wird klar, dass sie ein Konstrukt aus eigenen Kindheitserinnerungen, persönlichen Erfahrungen mit Kindern und dem kindlichen Gesellschaftsbild sind.“ (Liebal & Exner, 2011, S. 91)

Diese Argumentation spricht für die intensive Einbeziehung von Kindern, nicht nur in die abschließende Testphase, sondern bereits in die Analyse- und Konzeptionsphasen. (Liebal & Exner, 2011, S. 92)

5.2.1 Rollen

Kinder können im Design- bzw. Gestaltungsprozess verschiedene Rollen einnehmen. Nettet & Large (2004) klassifizieren diese Rollen nach ihrem Interaktionsgrad mit den Designerinnen und Designern:

1. Die Rolle der Benutzerin, des Benutzers ist die älteste der vier Design-Rollen. (Nettet & Large, 2004, S. 151) Hier findet keine aktive Einbeziehung der Benutzerin oder des Benutzers in den Gestaltungsprozess statt. (Markopoulos et al., 2008, S. 45)
2. Die zweite Rolle, die Kinder im Design-Prozess einnehmen können, ist die Rolle der Testerin, des Testers. Hierbei testen Kinder Prototypen der Technologie, bevor diese auf dem Markt eingeführt werden. Wie die Kinder mit den Prototypen interagieren, beeinflusst die Entwicklung zukünftiger Iterationen. Die Rolle der Testerin, des Testers hat ihre Wurzeln in Seymour Paperts LOGO-Forschungsgruppe am MIT in den 1980er Jahren. Die Testerrolle wurde in den 1990er Jahren im industriellen und akademischen Bereich sehr beliebt, sodass es heutzutage unüblich ist, wenn kommerzielle Produkte für Kinder nicht von Kindern getestet werden. (Nettet & Large, 2004, S. 151)

3. Die dritte Rolle ist die der Informantin, des Informanten und bildet die Grundlage des Informant Designs (siehe Kapitel 5.2.3.2). In dieser Rolle nehmen Kinder in verschiedenen Stadien am Design-Prozess teil, um Informationen für diesen zu liefern, wobei diese Teilnahme allerdings keine durchgehende Einbindung während der Entwicklung der Technologie darstellt. (Nesset & Large, 2004, S. 151)
4. Die vierte und letzte Rolle, die Kinder im Design-Prozess einnehmen können ist die der Gestaltungspartnerin oder des Gestaltungspartners, wie z.B. bei der Methode Cooperative-Inquiry (siehe Kapitel 5.2.3.3). Hier sind Kinder und Erwachsene durch den ganzen Design-Prozess hindurch gleichgestellte Partnerinnen und Partner. Im Gegensatz zu den Rollen Benutzerin, Benutzer, Testerin, Tester und Informantin, Informant ist bei der Rolle der Gestaltungspartnerin, des Gestaltungspartners die Wirkung der Kinder auf den Design-Prozess einer Technologie höher als die Wirkung, die die Technologie auf die Kinder hat. (Nesset & Large, 2004, S. 152)

Der Grad der Interaktion und der Anteil am Gestaltungsprozess der einzelnen Rollen kann grafisch wie folgt in einem Zwiebelschalenmodell abgebildet werden:

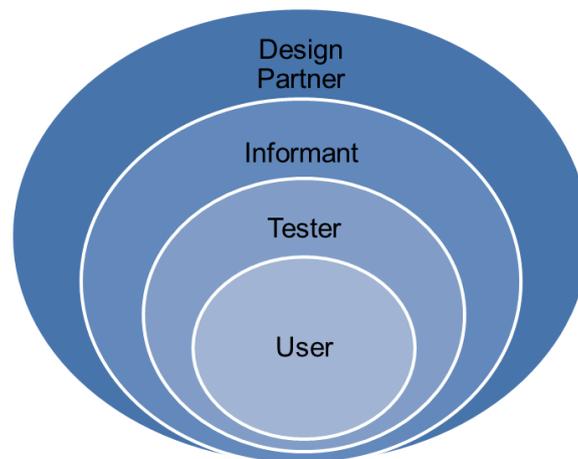


Abbildung 5: Interaktionsanteil am Gestaltungsprozess nach Rollen, Druin's Modell, eigene Darstellung nach (Markopoulos et al., 2008, S. 46)

Liebal & Exner (2011) klassifizieren diese Rollen entlang des klassischen Usability Engineering Lifecycles, der aus Anforderungsanalyse, Konzept, Prototyping und Testing besteht:

- | | |
|--------------------------------|---------------------------------------|
| 1. Phase: Anforderungsanalyse, | Rolle: Informantin, Informant |
| 2. Phase: Konzept, | Rolle: Benutzerin, Benutzer |
| 3. Phase: Prototyping, | Rolle: Gestaltungspartnerin, -partner |
| 4. Phase: Testing, | Rolle: Testerin, Tester |

(Liebal & Exner, 2011, S. 95)

5.2.2 User-Centered-Design, die Basis für Child-Centered-Design

Die aktive Integration der zukünftigen Benutzerinnen und Benutzer, vor allem auch die der Zielgruppe Kinder, in den Gestaltungsprozess von Soft- und Hardware hat sich in der Vergangenheit als sehr zielführend und gewinnbringend erwiesen.

„Most of the software that is intended for children is designed with very little input from children themselves.“ (Mazzone & Read, 2005, S. 1)

Um der Nichteinbeziehung von Benutzerinnen und Benutzern, im Speziellen von Kindern, in den Gestaltungsprozess von Produkten entgegenzuwirken, wurden Methoden entwickelt die auch bei der Zielgruppe Kinder gut eingesetzt werden können. (Mazzone & Read, 2005), (Joly, 2007) Child-Centered-Design (CCD) als Methode wird aus dem User-Centered-Design

abgeleitet und die User, also Benutzerinnen und Benutzer, sind in diesem Fall Kinder.²⁵ (Nesset & Large, 2004)

Zahlreiche Herangehensweisen an die Gestaltung der Human-Computer-Interaction verfolgen ähnliche Basiskonzepte. Der Grundgedanke: Die Benutzerin, der Benutzer weiß am besten, was sie, er mit einem Produkt machen will und wie sie, er es bedienen möchte. (Saffer, 2010, S. 33) Der User-Centered-Design-Ansatz selbst sieht eine „Problemdefinition“ vor, bei der der Kontext und die zukünftige Produktumgebung analysiert, sowie die zukünftigen Benutzerinnen und Benutzer identifiziert werden. Anschließend erfolgt eine detaillierte funktionale Spezifikation des Produkts, welches beim Testen durch Usability-Evaluierungs-Methoden auf seine Nutzbarkeit überprüft wird. In der benutzerzentrierten Anforderungsanalyse sind ebenso drei wesentliche Schritte vorgesehen: Zu Beginn findet eine „User Context Analysis“ statt, bei der das zu Grunde liegende Problem, die Zielgruppe und der Kontext für das zukünftige Produkt analysiert werden. Anschließend wird die Machbarkeit mittels Prototyping überprüft. Zum Schluss wird der Input der Benutzerinnen und Benutzer aus den ersten beiden Schritten in den Design-Prozess weitergegeben und so die Anforderungen an das Produkt zusammengeführt und überprüft. (Carter, 1999, S. 182) Im Gegensatz zum klassischen User-Centered-Design-Ansatz zeigen jedoch moderne Erkenntnisse, dass eine frühzeitige Einbeziehung von Benutzerinnen und Benutzer in den gesamten Gestaltungsprozess gewinnbringender ist. (Liebal & Exner, 2011, S. 92)

5.2.3 Weiterentwicklung des User-Centered-Designs für einen Einsatz mit Kindern als Gestaltungspartnerinnen und -partner

5.2.3.1 Participatory-Design

Im Participatory-Design (PD) ist die Einbindung der Benutzerinnen und Benutzer stärker als im Contextual-Design zu sehen. (Wook, Ashaari, Noor, Zin & Jaafar, 2011, S. 2) Es handelt sich um einen höchst iterativen Ansatz und die Methode verfügt über viele Freiheitsgrade und folgt kaum Standards. (Nesset & Large, 2004, S. 143)

Generell wird im Participatory-Design auf allgemein bekannte und einfache Methoden Wert gelegt, wie z.B. der Einsatz von Brainstorming und „Low-Tech Tools“, wie Tafeln, Indexkarten und Post-It-Notizen zur Visualisierung der Ideen. (Read, Gregory, MacFarlane, McManus, Gray & Patel, 2002, S. 53), (Walsh, Foss, Yip & Druin, 2013) Sluis-Thiescheffer (2007) zeigt auf, dass deutlich bessere Ergebnisse und ein höherer Mehrwert entstehen, wenn Prototyping intensiver genutzt wird, da Kinder mit manuellen Tätigkeiten schneller zu wertigen Ergebnissen kommen, als bei reiner geistigen Tätigkeit, wie es beim Brainstorming der Fall ist. (Sluis-Thiescheffer, 2007, S. 203), siehe auch (Yip, Foss, Bonsignore, Guha, Norooz, Rhodes, McNally, Papadatos, Golub & Druin, 2013)

Kritikerinnen und Kritiker der Methode vertreten die Meinung, Kinder seien nicht fähig als ernstzunehmende Partnerinnen und Partner im Design-Prozess zu agieren. (Nesset & Large, 2004, S. 144) Der Kritik entgegenstehen zahlreiche Case Studies und Forschungsberichte, die zeigen, dass Participatory-Design mit Kindern, sogar mit sehr jungen Kindern (im Alter von 3 bis 5 Jahren) durchgeführt werden kann. Siehe dazu u.a. (Seitinger, Sylvan, Zuckerman, Popovic & Zuckerman, 2006, S. 306) Participatory-Design kann auch mit 2- bis 7-jährigen Kindern eingesetzt werden. Tomitsch, Grechenig, Kappel & Költringer (2006) haben diese Methode in Kombination mit anderen eingesetzt, um gemeinsam mit Kindern ein „Tangible Music Toy“ zu gestalten und zu entwickeln. (Tomitsch et al., 2006, S. 170)

5.2.3.2 Informant-Design

Beim Informant-Design handelt es sich um eine Weiterentwicklung des User-Centered-Designs in Kombination mit dem Participatory-Design, mit besonderem Augenmerk auf die

²⁵ <http://uxkids.com/blog/child-centered-design-is-user-centered-design-but-then-different/>, 23.04.2018

Zielgruppe Kinder, wobei die Wurzeln dieses Design-Ansatzes in der Entwicklung von Bildungssoftware liegen. Beim Participatory-Design stellt sich die Gleichberechtigung der Kinder und Erwachsenen als praktische Herausforderung dar (auch bei Settings mit ausschließlich Erwachsenen). Um diesen Nachteil zu beseitigen, können beim Informant-Design Kinder als „Native Informants“, quasi als direkte Informantinnen und Informanten, agieren. (Nesset & Large, 2004, S. 147) In dieser Rolle liefern Kinder Informationen an die Designerinnen und Designer, agieren aber selbst nicht als solche. (Mazzone, Xu & Read, 2007, S. 153) Neben der Beobachtung der Kinder, liefern gerade diese Inputs in der Prototypingphase bei der Erstellung selbiger wichtige Informationen. Auch in diesem Modell sind Iterationen essentiell. (Nesset & Large, 2004, S. 147) Der Zeitpunkt der Einbindung der Kinder orientiert sich am tatsächlichen Design-Prozess (Walsh et al., 2013, S. 2894). Eine Einbindung von Kindern in unterschiedlichen Rollen, als Benutzerinnen und Benutzer, als Testerinnen und Tester, als Informantinnen und Informanten, als Design-Partnerinnen und -Partner, sogar als Forschungs-Mitarbeiterin bzw. -Mitarbeiter, hat sich in zahlreichen Studien als erfolgreich bewährt. (van Doorn, Stappers & Gielen, 2013, S. 2884), (Mazzone, Xu & Read, 2007, S. 153)

5.2.3.3 Cooperative-Inquiry und Bonded-Design

Basierend auf kooperativen Design-Praktiken („Cooperative-Design“) und teilnehmenden Design-Praktiken („Participatory-Design Practices“) ist das wesentliche Ergebnis von „Cooperative-Inquiry“ das Design-zentrierte Lernen („Design-Centered-Learning“). Als Methoden werden Feldstudien und iteratives Low- und High-Fidelity Prototyping eingesetzt, wobei Kinder als vollwertige Gestaltungspartnerinnen und -partner mitarbeiten. (Nesset & Large, 2004, S. 144)

Grundlegende Methodik ist, wie im Contextual-Inquiry, die Beobachtung der Benutzerinnen und Benutzer in ihrer eigenen Umgebung. (Nesset & Large, 2004, S. 144) Den Kindern wird in dieser Methode die Möglichkeit eingeräumt, aktiv bei der Gestaltung von Technologie mitzuwirken. (van Doorn, Stappers & Gielen, 2013, S. 2884), (Mazzone, Xu & Read, 2007, S. 153) Bei Kindern sollte eine erwachsene Person als teilnehmende Beobachterin, als teilnehmender Beobachter fungieren und zwei weitere Personen sollten Notizen machen. So kann die teilnehmende Beobachterin, der teilnehmende Beobachter frei mit den Kindern agieren und die Kinder ggf. aus der Reserve locken und sanfte Anleitung geben, falls benötigt. Kinder sind eher forschend als aufgabenorientiert aktiv, sofern man ihnen diese Freiheiten lässt. Es ist daher wichtig, dass Kindern diese Möglichkeiten geboten werden, denn dann entsprechen die gesammelten Daten wirklich dem, was die Kinder selbst denken und nicht dem, was die Kinder annehmen, dass von Seiten der Erwachsenen von ihnen erwartet wird. (Nesset & Large, 2004, S. 144)

Bei Prototyping-Aktivitäten ist darauf zu achten, dass Kinder gleichberechtigt eingebunden werden. Low-Fidelity-Prototyping (z.B. Paper-Prototyps) eignet sich auf Grund der Art der Aktivität und der verwendeten Materialien gut dazu. (Walsh et al., 2013, S. 2894) Die Arbeit mit Klebe-Notizzetteln, die flexibel an die jeweiligen Artefakte angeheftet werden können, unterstützt die iterative Herangehensweise, da die Notizen auch ggf. wieder leicht entfernt werden können. (Yip et al., 2013, S. 293)

Beim „Bonded-Design“, das sehr ähnlich zum Cooperative-Inquiry abläuft, ist die Zusammenarbeit der Kinder mit den Forscherinnen und Forschern kürzer und die Aktivitäten werden in Schulen statt in Labor-Umgebungen durchgeführt. Alle Teilnehmerinnen und Teilnehmer am Prozess werden sowohl als Lernende (Benutzerinnen und Benutzer des Lernprodukts) als auch als Designerinnen und Designer angesehen. (Walsh et al. 2013, S. 2894)

Eine weitere Variante des Cooperative-Inquiry stellt das sog. Curriculum-Focused-Design dar. Hierbei werden die Design-Aktivitäten in das reguläre Curriculum bzw. in den Schulunterricht integriert. Wertvoll für den Design-Prozess erweisen sich die natürlichen Kooperationen der Kinder im Klassenkontext, welche im Gegensatz zum Laborsetting, mit

Kindern, die sich noch nicht kennen, schon existieren. Besonders hilfreich ist es für beide Seiten, für die Designerinnen und Designer sowie für die Schule, wenn in den Design-Aktivitäten curriculare Themen behandelt werden, wie z.B. die Gestaltung eines bestimmten Geräts, welches im Unterricht vorkommt. (Rode, Stringer, Toye, Simpson & Blackwell, 2003, S. 125)

5.2.4 Card Sorting mit Kindergartenkindern

Joly, Pemberton & Griffiths (2009) erprobten und evaluierten den Einsatz von „Card Sorting“-Verfahren mit Kindergartenkindern. Sie führen an, dass „Card Sorting“ bereits mit 8-jährigen Kindern erfolgreich durchgeführt wurde und sie mit Hilfe von „Card Sorting“-Experimenten mit 3- bis 4-jährigen Kindern testen, ob die Methode auch mit dieser Zielgruppe sinnvoll eingesetzt werden kann. (Joly, Pemberton & Griffiths, 2009, S. 204)

Sie unterscheiden dabei drei Arten von „Card Sorting“-Verfahren: Open Card Sorting, Closed Card Sorting und Match-To-Sample Card Sorting.

Das klassische Card Sorting beschäftigt sich mit der Aufgabe, dass Probandinnen und Probanden Karten mit darauf vorgeschlagenen Konzepten in Gruppen sortieren sollen. Häufigster Anwendungsfall hierfür ist die Gestaltung von Menüs und Befehlssets für grafische Benutzerschnittstellen. Die Probandinnen und Probanden werden gebeten, die Karten in Gruppen zu unterteilen und werden im Anschluss befragt, nach welchen Kriterien sie die Sortierung der Karten vorgenommen haben. (Joly, Pemberton & Griffiths, 2009, S. 204)

Bei einem „Open Card Sorting“ sind die Probandinnen und Probanden an keinerlei Vorgaben gebunden, außer an die Konzepte, die auf den Karten repräsentiert werden. Sie müssen selbst entscheiden, wie viele Gruppen, Kategorien und welche Reihenfolge der Karten sie anlegen möchten. Auch die Benennung der Gruppen, Kategorien und Überschriften / Beschriftungen ist ihnen hierbei überlassen. (Joly, Pemberton & Griffiths, 2009, S. 204 & S. 207)

Im Gegensatz dazu werden beim „Closed Card Sorting“ oftmals Vorgaben gemacht. Solche Vorgaben können z.B. Überschriften oder einzelne Gruppen sein. Die Probandinnen und Probanden sollen dann entweder die Reihenfolge der Beschriftungen festlegen oder die vorhandenen Elemente in die jeweiligen Gruppen einsortieren. (Joly, Pemberton & Griffiths, 2009, S. 204 f), (Nielsen, 1993, S. 127)

Zur Verfeinerung des „Closed Card Sorting“ setzten Joly, Pemberton & Griffiths (2009) das „Match-To-Sample“-Verfahren ein, in dem statt den einzelnen Kategorie-Bezeichnern einige Vertreter-Karten der jeweiligen Kategorien gezeigt werden und neue Karten diesen zugeordnet werden müssen. Durch diese Vorgehensweise soll festgestellt werden, ob die Kategorien tatsächlich aussagekräftig gewählt wurden. (Joly, Pemberton & Griffiths, 2009, S. 209)

Am Test von Joly, Pemberton & Griffiths (2009) nahmen 56 Kinder, davon 27 Mädchen und 29 Jungen, teil. In einer ersten Phase fand ein „Closed Card Sorting“ und in einer zweiten Phase ein „Open Card Sorting“ statt. (Joly, Pemberton & Griffiths, 2009, S. 205 & S 208) Die Tests ergaben, dass „Closed Card Sorting“-Verfahren mit Kindergartenkindern erfolgreich durchgeführt werden können, solange sie als „Spiel“ getarnt sind und sich mit Themen, die Kinder interessiert, beschäftigen. „Open Card Sorting“-Verfahren benötigen mehr Zeit und müssen nach den Erkenntnissen von Joly, Pemberton & Griffiths (2009) noch weiterentwickelt werden. Die Kombination des „Match-To-Sample“ mit dem „Closed Card Sorting“ erachten sie als sehr hilfreich, wenn Kinder als Informantinnen und Informanten im Design-Prozess teilnehmen. So können die Kategorien dahingehend definiert und verfeinert werden, dass sie am besten die Auswahl der Kinder widerspiegeln. (Joly, Pemberton & Griffiths, 2009, S. 211) Darüber hinaus konnten Joly, Pemberton & Griffiths (2009) keinen Zusammenhang zwischen dem Geschlecht der Kinder und ihrer Fähigkeit Kategorisierungen vorzunehmen feststellen. (Joly, Pemberton & Griffiths, 2009, S. 206)

5.3 Usability Evaluation mit Kindern im Alter von 3 bis 6 Jahren

Joly (2007) weist auf das Ergebnis früherer Untersuchungen hin, wonach es in Hinblick auf die Testergebnisse einer Usability-Studie mit Kindern keine Rolle spielt, ob ein Prototyp, ein Wizard of Oz oder das fertige lauffähige Produkt eingesetzt wird. (Joly, 2007, S. 187)

Um die Benutzerfreundlichkeit von Systemen für Kinder evaluieren zu können, gilt es geeignete Methoden einzusetzen. Generell ist es empfehlenswert die Evaluierungen bzw. die Usability-Tests aufzuzeichnen, um die Interaktion und das Verhalten, wie Lächeln, Lachen, Stirnrunzeln und Seufzer, im Anschluss auswerten zu können. Diese Beobachtungen der Gestik und Mimik helfen bei der Auswertung und sind oftmals aussagekräftiger, als die Antworten der Kinder auf Fragebögen, dahingehend ob sie etwas mögen oder nicht. (Joly, 2007, S. 187)

Im Folgenden stellt die Autorin exemplarische Methoden vor, die mit der Zielgruppe Kinder eingesetzt werden können.

5.3.1 Thinking und Talk Aloud

Eine leicht und schnell einsetzbare Methode ist das Thinking Aloud. Dabei reden die Probandinnen und Probanden während des Tests laut mit und erläutern, was sie denken. (Xu, Mazzone & MacFarlane, 2006, S. 171)

„Thinking Aloud (TA) provides the evaluator with information about the cognition and emotion of a user while the child performs a task or solves a problem.“ (Xu et al., 2007, S. 1012)

Die Methode ist für Kinder jedoch ziemlich fordernd. Man könnte daher nach dem durchgeführten Test einen Fragebogen ausfüllen lassen (sofern Schreibkompetenzen vorhanden sind) oder ein Interview führen (d.h. eine Retrospektion durchführen) (van Kesteren, Bekker, Vermeeren & Lloyd, 2003, S. 42), was jedoch den Nachteil mit sich bringt, dass die Probandinnen und Probanden oftmals schon Details vergessen haben. (Baauw & Markopoulos, 2004, S. 115) Da die Methode für Kinder nicht nur fordernd, sondern oftmals das laute Denken durch die Anwesenheit der zumeist erwachsenen Testleiterin, des erwachsenen Testleiters beeinflusst wird, haben Markopoulos, Verschoor, IJsselsteijn & de Ruyter (2008b) einen „Interventions-Roboter“ entwickelt. Dieser Roboter, hier eine Philips iCat, erfasst das von den Kindern „laut Gedachte“ und die Testleiterinnen und Testleiter können die Ergebnisse anschließend auswerten. (Markopoulos et al., 2008b, S. 118 f.)

Eine Abwandlung von Thinking Aloud ist Talk Aloud:

„Talk aloud is an adaptation of think-aloud where children are instructed to talk about what they are doing instead of what they are thinking.“ (Donker & Reitsma, 2004, S. 43)

Das laute Beschreiben von Aktionen fällt nicht allen Kindern leicht und muss von den Testleiterinnen und -leitern immer wieder eingefordert werden. Es gilt hierbei zu berücksichtigen, dass Kinder gerne ihre Emotionen und generellen Empfindungen in diese Beschreibungen packen. Diese Informationen können in weiterer Folge zur Beurteilung der Begeisterung und Einstellung zur Nutzung des Produkts herangezogen werden. (Donker & Reitsma, 2004, S. 48)

5.3.2 Heuristische Evaluation

Eine sehr kostengünstige und effiziente Usability Evaluations-Methode ist die heuristische Evaluation, die maßgeblich von Nielsen & Molich (1990) geprägt wurde. Um die Qualität und Quantität der Erkenntnisse aus der heuristischen Evaluation zu steigern, können statt Endbenutzerinnen und -benutzern des Produkts Spezialistinnen und Spezialisten aus dem Bereich User-Interface-Design und Usability eingesetzt werden. (Nielsen, 1992, S. 377)

Als sehr leichtgewichtige und informelle Methode, und damit als erste Stufe einer Usability Evaluation, kann eine Usability Inspektion angesetzt werden. Diese ähnelt in ihrem Vorgehen einer Heuristischen Evaluation, auch die Überprüfung in Bezug auf Guidelines und

Heuristiken ist ihr gleich. (Rester, Pohl, Banovic, Hinum, Miksch, Ohmann & Popow, 2006, S. 144) Im Kontext von Software zum Lernen haben Squires & Preece (1999) in Anlehnung an die Usability Heuristiken von Nielsen, „Learning with Software“-Heuristiken entwickelt. (Squires & Preece, 1999, S. 467) Hintergrund ist die Entwicklung einer Evaluationsgrundlage für Lehrkräfte, um geeignete Software für den Unterricht auswählen zu können. Dabei soll neben pädagogischen Prinzipien (kognitive Authentizität und Kontextauthentizität) auch auf die Usability besonders Rücksicht genommen werden. (Squires & Preece, 1999, S. 468 f.) In den Evaluationskriterien wird unter anderem auf Aspekte, wie die Übereinstimmung der Intentionen der Designerinnen und Designer mit dem Lernmodell bzw. den Bedürfnissen der Lernenden (Squires & Preece, 1999, S. 479) oder Navigierbarkeit und angemessene Ebenen an Lernkontrollen (Squires & Preece, 1999, S. 480), eingegangen. Weitere Aspekte sind die Übereinstimmung mit dem Curriculum, die gewählten symbolischen Repräsentationen oder auch die Fähigkeiten der Software periphere kognitive Fehler präventiv zu unterbinden. (Squires & Preece, 1999, S. 480)

Alsumait & Al-Osaimi (2009) haben die Heuristiken von Nielsen & Molich (1990) um zwei weitere Kategorien, „Child Usability Heuristics“ und „E-Learning Usability Heuristics“ ergänzt, um damit E-Learning-Produkte für Kinder zu evaluieren und zu testen. (Alsumait & Al-Osaimi, 2009, S. 427 f.)

5.3.3 Prototyping mit Kindern

„Aside from the finished product, prototypes are the ultimate expression of the interaction designer’s vision“ (Saffer, 2010, S. 174)

Eine Einbeziehung von Kindern im Gestaltungsprozess von Produkten kann hauptsächlich über Prototyping geschehen. (Liebal & Exner, 2011, S. 191) Im Participatory-Design (siehe Kapitel 5.2.3.1) haben Kinder die Möglichkeit als aktive und gleichberechtigte Designerinnen und Designer mitzuwirken. Wobei aufgrund der fehlenden Kompetenzen der Kinder nach Ansicht von Liebal & Exner (2011) ausschließlich Low-Fidelity Prototypen erstellt werden können. (Liebal & Exner, 2011, S. 192)

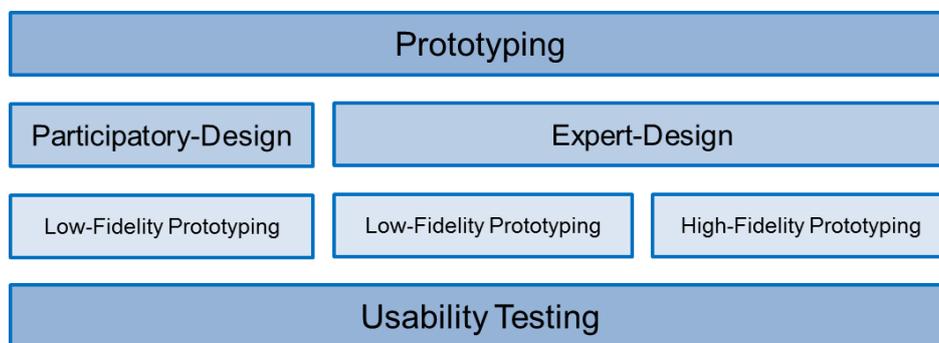


Abbildung 6: Einsatz von Prototyping, eigene Darstellung in gekürzter Form nach (Liebal & Exner, 2011, S. 191)

„Low-Tech-Prototypen [Low-Fidelity Prototypen, Anm. d. Verf.] werden mit Hilfe verschiedenster Materialien durch Kinder und Erwachsene gleichsam hergestellt und dienen der Entwicklung neuer Ideen. Diese Technik [Participatory-Design, Anm. d. Verf.] kann mit Kindern ab drei Jahren durchgeführt werden, ist aber mit Kindern im Alter von sieben bis zehn Jahren am effektivsten.“ (Liebal & Exner, 2011, S. 197)

Low-Fidelity Prototypen können universell eingesetzt werden: Sie eignen sich zum Testen und Gestalten von Desktop-Benutzerschnittstellen, für Apps oder aber auch für Tangible Interfaces, da sie auf Grund ihrer Eigenschaften einfach zu erstellen, flexibel und intuitiv zu bearbeiten sind. (Rick, Fleck, Yuill, Francois, Fields & Carr, 2010, S. 143) Gerade bei Tangible Interfaces für Kinder ist umfangreiches Prototyping unumgänglich, da nicht alle Charakteristika eines Tangible Interfaces in nur einem oder zwei Prototypen abgedeckt

werden können. (Xu, Read, Mazzone & Brown, 2007, S. 28) Ähnliches berichten Kussmaul & Jack (2006, S. 189).

Prototyping mit Kindern verlangt nach Erwachsenen, die diesen Prozess intensiv begleiten und moderieren, aber Kindern die Freiheiten zur Gestaltung gestatten. (Liebal & Exner, 2011, S. 197) Zur Evaluation und beim Testen von Prototypen durch Kinder ist zu beachten, dass diese eher explorativ Benutzeroberflächen und Produkte erkunden. Bei horizontalen Prototypen, die einen hohen Funktionsumfang suggerieren, aber kaum Funktionalität besitzen, können Kinder leicht das Interesse verlieren, da sie hinter den jeweiligen Interaktionselementen keine neuen Entdeckungen machen können. Rein horizontale Prototypen sind daher nur bedingt für Tests mit Kindern geeignet. (Liebal & Exner, 2011, S. 196)

Das Erstellen von Zeichnungen durch Kinder im Alter von 4 bis 5 Jahren ist eine sehr gute Möglichkeit zur unterstützenden Gestaltung von Benutzerschnittstellen. Kinder können nach der Nutzung einer Benutzerschnittstelle ihre Erlebnisse bildhaft darstellen. Diese Bilder können vor allem zur Interpretation von Spaß und Engagement der Kinder bei der Benutzung herangezogen werden, weniger jedoch zur Evaluation der Benutzerfreundlichkeit, wie Sylla (2010) aufzeigt. (Sylla, 2010, S. 4385)

„Drawing Intervention (DI) is new and more of an informal evaluation method, inspired by the natural school classroom settings. According to it, after the children completed the tasks, we let them sit around a table, and draw anything related to what they have done, and also write a sentence about their drawings.“ (Xu et al., 2007, S. 1012)

In der Praxis besteht die Möglichkeit, fertige Low-Fidelity-Interaktionselemente, wie z.B. auf Karton vorgefertigte Icons oder Bilder einzusetzen, die die Kinder anschließend auf einem Blatt Papier oder beliebigem Hintergrund anordnen und so einen ersten Entwurf einer Benutzerschnittstelle selbst gestalten können. (Scaife & Rogers, 1998, S. 36)

5.4 Weitere Forschungsergebnisse und Best-Practice-Beispiele

Die Verfasserin der gegenständlichen Arbeit stellt in diesem Kapitel exemplarisch Projekte vor, bei welchen neue Produkte für Kinder im Alter von 3- bis 6 Jahren in Zusammenarbeit mit Kindern der jeweiligen Zielgruppe entwickelt wurden. Besonderes Augenmerk legt die Autorin auf Hinweise zur Prozessdurchführung und darauf welchen Beitrag Kinder im Alter von 3 bis 6 Jahren zu leisten im Stande sind.

Die Arbeit von Pane (1997), Pane (2001), Pane, Myers & Miller (2002) beschäftigt sich unter anderem damit, dass fortwährende Usability-Evaluierungen mit der Zielgruppe, in diesem Fall Kinder ab 10 Jahren (auch wenn das nicht die Zielgruppe der gegenständlichen Arbeit ist), für eine hohe Benutzerfreundlichkeit unabdingbar sind. Mit HANDS (Human-centered Advances for the Novice Development of Software) haben die Autoren eine grafische, event-basierte Programmiersprache und zeitgleich auch eine geeignete Programmierumgebung für Kinder entwickelt, die an der natürlichen Sprache der Benutzerinnen und Benutzer angelehnt ist. Die Entscheidung, ob eine text- oder grafikbasierte Programmierumgebung eingesetzt werden sollte, fiel zugunsten eines hybriden Ansatzes, da Pane, Myers & Miller (2002) keine klaren Studien darüber finden konnten, ob textbasierte oder rein grafische Programmierumgebungen benutzerfreundlicher sind. In eigenen Studien konnten sie beobachten, dass die Untersuchungsteilnehmerinnen und -teilnehmer je nach Art der Programmieraufgabe zu text- oder grafikunterstützten Hilfsmitteln griffen, diese also, wie im hybriden Ansatz gedacht, kombinieren. Mit grafischen Elementen wurde die Programmstruktur erstellt, Aktionen und Eigenschaften wurden mittels Text hinzugefügt. (Pane, Myers & Miller, 2002, S. 200)

Mit der Entwicklung des sog. „Sneetches Room“ verfolgten Montemayor, Druin, Farber, Simms, Churaman & D'Amour (2002) einen „physical programming“-Ansatz, bei dem Kinder in „physical interactive environments“ mit Gegenständen Szenen programmieren können und so in der Lage sind Geschichten zu erzählen. (Montemayor et al., 2002, S. 300 f.) In diesem

Entwicklungsszenario wirkten die Kinder von Anbeginn als Partnerinnen und Partner beim Design mit und unterstützten aktiv bei der Gestaltung von Skizzen, Low-Fidelity-Prototypen, Walkthrough-Szenarien oder auch Wizard of Oz Prototypen. (Montemayor et al., 2002, S. 301) Hier zeigte sich ganz klar, dass auch junge Kinder im Alter von 4 bis 6 Jahren in der Lage sind aktiv am Gestaltungsprozess teilzunehmen. (Montemayor et al., 2002, S. 304)

Cairncross & Waugh (2005) haben sich mit Child-Centered-Design für die Zielgruppe der 3- und 4-jährigen Kinder auseinandergesetzt. (Cairncross & Waugh, 2005, S. 1119) Sie zeigen in ihrer Arbeit auf, dass es bei jungen Kindern deutlich mehr an Moderation bedarf, als bei älteren Kindern, um diese Methoden durchzuführen. Die gemeinsamen Gestaltungsaktivitäten erfordern von den Designerinnen und Designern ein Verständnis über die kognitive Entwicklung der Kinder in diesem Alter. (Cairncross & Waugh, 2005, S. 1122 f.) Durch die zahlreichen Aktivitäten mit Kindern werden Anforderungen erkannt und erhoben, die sonst nicht bekannt sein würden. Die Kinder leisten somit einen wertvollen Beitrag zur Gestaltung von Produkten und Benutzeroberflächen. Darüber hinaus ist es Cairncross & Waugh (2005) ein Anliegen, dass für ein adäquates Design, neben jungen Kindern, auch andere Stakeholder und Expertinnen und Experten in den Design-Prozess eingebunden werden. (Cairncross & Waugh, 2005, S. 1123) Zusätzlich zeigt Antle (2007) auf, dass Kinder bei der Bedienung von Technologie gerne in ihre natürliche Sprache und Verhaltensmuster zurückwechseln, wenn die Technologie nicht unmittelbar das tut, was sie sich vorstellen. (Antle, 2007, S. 201)

deDiego-Cottinelli & Barros stellten 2010 eine prototypische Applikation für Tablet-PCs vor (TRAZO), die basierend auf einem Anforderungsprofil von Lehrkräften 3-jährigen Kindern erste Vorübungen zum Schreibenlernen (Schreibschrift mit der Hand) anbietet. Neben den umfangreichen Auswertungsmöglichkeiten der Eingaben der Kinder (in Hinblick auf Speichern und Vergleichen der Lösungen, Geschwindigkeit, etc.), die sich durch den Einsatz eines Tablet-PCs anstelle von Papier leicht vornehmen lassen, zeigte sich in den Evaluationen, dass 3-jährige Kinder mit dem angebotenen Programm auf einem Tablet-PC sehr gut zurechtkommen und die Bedienung des Tablet-PCs schnell erlernten. (deDiego-Cottinelli & Barros, 2010, S. 281)

Kindborg & McGee (2007) haben mit Comikit eine grafische Programmierumgebung für Kinder iterativ mit Hilfe von Prototypen entwickelt. Bei der Entwicklung kamen in den Feldstudien mit 9- bis 11-jährigen Kindern sowohl Low-Fidelity als auch High-Fidelity Prototypen zum Einsatz. (Kindborg & McGee, 2007, S. 111) Als größte Herausforderung in der Gestaltung von Comikit stellte sich die Auswahl der Symbole dar. Während es für Charaktere und Objekte ideale Repräsentationen in Comics gibt, können abstrakte Kontrollstrukturen schwer mit bekannten Comic-Elementen repräsentiert werden. Es mussten neue Symbole entwickelt und getestet werden. (Kindborg & McGee, 2007, S. 117 f.)

Auch wenn die Zielgruppe von Kindborg & McGee (2007) 9- bis 11-jährige Kinder umfasst, und nicht wie die gegenständliche Arbeit 3- bis 6-jährige, ist eine wesentliche Erkenntnis aus deren Arbeit, dass die Generalisierbarkeit der Symbolik im Kontext der Zielgruppe Kinder nicht trivial ist. Symbole müssen universell einsetzbar sein und zugleich die Bedeutung von Kontrollstrukturen einer Programmierung repräsentieren. (Kindborg & McGee, 2007, S. 118) Darüber hinaus muss die Bedeutung der Symbole im jeweiligen Kontext berücksichtigt werden. (Kindborg & McGee, 2007, S. 123)

6 Forschungslücken und -bedarf

Die Autorin zeigt im folgenden Kapitel die bestehenden Forschungslücken im Themenumfeld der gegenständlichen Arbeit auf, die durch Einsatz von ihr eigens entwickelten Kriterienkatalogen von ihr bestimmt wurden.

Im ersten Abschnitt (siehe Kapitel 6.1) präsentiert die Verfasserin der gegenständlichen Arbeit exemplarische Forschungsprojekte, die sich mit IKT und Technologien als Lehr-Inhalt im Bildungsbereich, mit Fokus auf Kindergartenkinder und frühe Volksschule, beschäftigen und zeigt die, entlang eines Kriterienkatalogs belegte, Forschungslücke auf.

Darauffolgend (siehe Kapitel 6.2) stellt die Autorin aktuelle wissenschaftliche Publikationen zu den Themen „Computational Thinking“, „Educational Robotics“ und „Robotics in Education“ vor und bestimmt die bestehende Forschungslücke anhand eines Kriterienkatalogs.

Der dritte Abschnitt (siehe Kapitel 6.3) betrachtet die bestehende Forschungslücke, wiederum begleitet von einem Kriterienkatalog, im Themenbereich des Einsatzes eines Child-Centered-Design-Ansatzes zur frühen Technikbildung in Informatik.

Abgerundet wird das Kapitel (siehe Kapitel 6.4) mit einem allgemeinen, kurzen Zwischenfazit zur frühen Technikbildung in Informatik.

6.1 Research Gap: Einsatz von IKT im Bildungsbereich

Im Bereich des Einsatzes von IKT im Bildungsbereich – vornehmlich im Kindergarten – gilt es eine Forschungslücke zu schließen, wie die folgende Darstellung zeigt. Gestützt wird die Argumentation der Forschungslücke durch folgende Kriterien zur Beurteilung der Projekte und Publikationen:

- Die jeweilige Forschungsarbeit behandelt als Zielgruppe Kindergartenkinder im Alter von 3 bis 6 Jahren.
- In der jeweiligen Forschungsarbeit werden konkrete Aufgaben vorgestellt, die direkt in den Lehr-Lern-Alltag übernommen und idealerweise auch variiert werden können.
- Die in der jeweiligen Forschungsarbeit vorgestellten Aufgaben und Aktivitäten unterstützen zuvor definierte Lernergebnisse.
- In der jeweiligen Forschungsarbeit wird eine mögliche Förderung von Kompetenzen nachvollziehbar und überprüfbar dargelegt.
- Die in den jeweiligen Forschungsarbeiten behandelten Themen, die in Lehr-Lern-Szenarien vermittelt werden sollen, beschäftigen sich mit Informatik im Speziellen oder Technologien im Allgemeinen und könnten einer frühen Technikbildung in Informatik dienlich sein.

Im 6. Rahmenprogramm der Europäischen Kommission werden im Projekt UPDATE (Understanding & Providing a Developmental Approach to Technology Education) unter anderem Best-Practice-Beispiele zur Integration von Technik-Themen in die elementare Bildung von Kindern erarbeitet und vorgestellt. Aus drei Blickwinkeln (Produzentin und Produzent, Benutzerin und Benutzer oder Wartungsperson) soll es Kindern im Alter von 4 bis 6 Jahren ermöglicht werden, durch Spiel und aktives Arbeiten mit Technologien bzw. Technik die vier Stufen des Lernkreises zu erfahren: Erleben und Erkenntnis, Erforschung, Untersuchung, Anwendung und neue Erkenntnis. (Turja, 2008, S. 6)

Im Rahmen des UPDATE-Projektes erfolgt die Entwicklung mehrerer Workshop-Szenarien, die sich nicht nur mit IKT, sondern auch mit Elektrizität, Holzbearbeitung und Basteln im Allgemeinen beschäftigen. In einem dieser Projekte, mit dem Fokus auf Robotern, arbeiten je eine Studierende bzw. ein Studierender mit zwei Kindern zusammen am Bau eines Roboters. Hierbei nehmen die Kinder die Rolle der Produzentinnen oder Produzenten ein und versuchen im Rahmen eines Spiels – das Roboter-Design ist missglückt und nun bedarf

es der Hilfe der Besucherinnen und Besucher des Roboterlabors um dieses zu verbessern – gemeinsam Roboter zu basteln. (Turja, 2008, S. 11 ff.)

Im Anhang der Publikation von Turja (2008) werden fünf Projekte für die Zielgruppe näher vorgestellt. Neben der Nennung der zu erzielenden Lernergebnisse ist eine detaillierte Beschreibung der Aktivitäten enthalten – diese sind jedoch nur mit zum Teil erheblichem Aufwand umsetzbar und es bedarf, je nach Projekt, auch der Unterstützung von Partnerinstitutionen. Leider fehlt eine kompetenzbasierte Darstellung und Verortung der Aktivitäten sowie eine Schwerpunktsetzung auf Informatik.

Mit dem „Early Childhood Education Information Communication Technology Professional Learning Programme (ECE ICT PL)“ hat das neuseeländische Bildungsministerium ein Forschungsprojekt durchgeführt (2005 bis 2009), an dem bis zu 70 Kinderzentren (Kindergärten und Kindertagesstätten) teilgenommen haben. Hierbei geht es primär darum IKT in den Alltag zum Spielen und Lernen zu integrieren, die Entwicklung der Kinder im Zusammenhang mit IKT zu beobachten und die entsprechenden Erkenntnisse daraus abzuleiten. Die zentralen Ergebnisse: IKT bereichert das Lernverhalten und Lernen an sich, die Qualität der Lehre wird gesteigert und es zeigt sich klar, dass es weniger eine Rolle spielt, welche Technologien zum Einsatz kommen, als wie und wozu Technologien genutzt werden. (Fagan, 2011, S. 2955) Darüber hinaus wird klar hervorgehoben, dass die Pädagoginnen und Pädagogen eine sehr große Rolle spielen und notwendig sind für einen zielbringenden Einsatz von IKT in der Bildung. (Fagan, 2011, S. 2959) Fagan (2011) verzichtet jedoch darauf die eingesetzten Aktivitäten, zu erzielende Lernergebnisse oder zu erreichende Kompetenzen vorzustellen.

Basierend auf einem Set an pädagogischen Prinzipien setzen Rivera et al. (2002) Technologien in der Lehre ein. CHAT (Cultural-Historical-Activity Theory) hat u.a. Vygotskys Theorie der „Zone of proximal development“ (siehe Kapitel 2.1.5), sowie den Konstruktivismus als Grundlage. (Rivera et al., 2002, S. 182) Bei CHAT wird davon ausgegangen, dass jegliche Interaktion der Lernenden mit Materialien und Technologien in einem sozialen Kontext stattfindet. (Rivera et al., 2002, S. 183) Als Beispiele führen Rivera et al. (2002) ein naturwissenschaftliches und ein technisches Szenario an. Zum einen werden Pflanzensamen gesät und gegossen und diese Aktivität in sechs Arbeitsschritte zerlegt, welche mit den Kindern besprochen werden. (Rivera et al., 2002, S. 194 ff.) Zum anderen werden die Tätigkeiten der digitalen Fotografie unter die Lupe genommen und den Kindern in Zusammenarbeit mit den Pädagoginnen und Pädagogen die einzelnen Schritte vom Erstellen eines Fotos, über das Übertragen desselben auf den PC bis hin zur Bildbearbeitung und zum Druck des Fotos nähergebracht. (Rivera et al., 2002, S. 199 ff.) Weiterführende Beispiele, die Darlegung von Lernergebnissen oder von Kompetenzen fehlen. Eine Sammlung von zahlreichen Aktivitäten in Kombination mit den zu erzielenden Lernergebnissen ist in Entz & Galarza (2000) zu finden, jedoch beschränken sich die darin enthaltenen Aufgabenstellungen auf den Einsatz von Digitalkameras als Technologie-Unterstützung zum Lernen.

In Österreich gibt es neben der Robocup Junior Initiative, die allerdings erst für Kinder ab 10 Jahren gedacht ist, weitere Projektinitiativen zur frühen Technikbildung. Mit dem Förderprogramm „Talente entdecken: Nachwuchs – Talente regional“ fördert die Österreichische Forschungsgesellschaft Projekte, „die Kindern und Jugendlichen ermöglichen, sich über einen längeren Zeitraum mit Forschung, Technologie und Innovation (FTI) in den Bereichen Naturwissenschaft und Technik auseinanderzusetzen.“²⁶

Mit diesem Förderprogramm wurde ein fruchtbarer Boden für neue Ideen und innovative Projekte geschaffen, die Kinder an Technologien heranzuführen. Im Kontext der

²⁶ <https://www.ffg.at/talente-entdecken-nachwuchs-talente-regional>, 23.04.2018

gegenständlichen Arbeit sind exemplarisch die Projekte ITAKE, READY und WIZIK anzuführen.

Einzig das Projekt ITAKE umfasst als Zielgruppe auch Kinder im Kindergartenalter, wobei anzumerken ist, dass keine wissenschaftlichen Publikationen, Abhandlungen, Studien oder Aufgabenstellungen zu diesem Projekt gefunden werden konnten. Das Projekt ITAKE hatte als Ziel Bewusstseinsbildung für die Wichtigkeit von IT als Lehrinhalt zu betreiben und hat so einige (ein bis drei Unterrichtseinheiten dauernde) Veranstaltungen mit Bodenrobotern an Kindergärten und Volksschulen ermöglicht. Einige Vernetzungsveranstaltungen unterstrichen die Bedeutung der Informatik und Technologien in der frühkindlichen Bildung.²⁷

Als Nachfolgeprojekt von ITAKE wurde READY durchgeführt. Ziel dieses Projekts war es ein „schulstufenübergreifendes, konsistentes und gendergerechtes Curriculum von der Volksschule bis zur Matura im Bereich ‚Messen, Steuern, Regeln‘ zu entwickeln und zu testen.“²⁸ Durch das Projekt READY ist die Möglichkeit entstanden, dass sich teilnehmende Schülerinnen und Schüler zum „OCG Certified Junior Robotics Engineer“ zertifizieren lassen können. (Rubenzer, Richter & Hofmann, 2015) In der Publikation, deren Zielgruppe Schulkinder und nicht 3- bis 6-jährige Kinder sind, werden nur kleine Auszüge an Lernzielen vorgestellt, aber keine konkreten Aufgaben und auch keine belegte mögliche Förderung von Kompetenzen.

Die Projektinitiative WIZIK (Wiener Zauberschule der Informatik) entwickelte einen sog. Zauberkoffer, der unterschiedliche Materialien und Aufgabenstellungen zur Vermittlung informatischer Grundkonzepte beinhaltet. Diese informatischen Grundkonzepte umfassen unter anderem das Binärsystem oder einen ersten Einblick in logisches und prozessorientiertes Denken. Dazu wurden auch Methoden des „Computer Science Unplugged“ Projekts (siehe Kapitel 3.1.1) verwendet. Dieser Zauberkoffer ist für einen Einsatz in der Volksschule, was nicht der Zielgruppe der gegenständlichen Arbeit entspricht, gedacht.²⁹ (Stockinger & Futschek, 2016) Mit welchen Lernergebnissen die im Zauberkoffer enthaltenen Aufgaben versehen und belegt sind, sowie deren möglicher Beitrag zur Kompetenzförderung ist, nach Recherchen der Verfasserin der gegenständlichen Arbeit, nicht publiziert.

Schachtner, Roth & Frankl (2005) erarbeiteten ein „mediendidaktisches-pädagogisches Konzept zum Einsatz des Computers im Vorschulalter“, mit dem „das Medium Computer für eine ganzheitliche Förderung der kindlichen Persönlichkeit zu nutzen“ ist und „Medienkompetenz als neue Kulturtechnik“ vermittelt werden soll. (Schachtner, Roth & Frankl, 2005, S. 41) Dieses Konzept behandelt nicht Informatik, nicht Computational Thinking und nicht Programmierung, sondern stellt exemplarische Aktivitäten zur Förderung der Medienkompetenzen vor.

Im Folgenden werden zwei exemplarische internationale Projekte zur Vermittlung von Grundkenntnissen in der Programmierung bzw. Informatik erwähnt. CoderDojo bietet zur Freizeitbeschäftigung „Programmierclubs“ allerdings erst für 7- bis 17-jährige Kinder an. Im Rahmen von Workshops und informellen Settings können Kinder erste Programmiererfahrungen sammeln, Webseiten erstellen und vieles mehr. Im Vordergrund steht der soziale und kreative Aspekt der Informatik, der durch CoderDojos gefördert werden soll. CoderDojos können von jeder und jedem auf der Welt gegründet werden und alle CoderDojos sind in einer großen Community miteinander vernetzt.³⁰ CoderDojos werden zwar in wissenschaftlichen Publikationen immer wieder erwähnt, aber stets nur als

²⁷ <http://www.ocg.at/de/itake-workshops>, 23.04.2018; <http://www.ocg.at/de/itake>, 23.04.2018, http://blog.ocg.at/2013/07/itake_symposium/, 23.04.2018

²⁸ https://www.ffg.at/sites/default/files/allgemeine_downloads/strukturprogramme/talreg_ready.pdf. 23.04.2018

²⁹ <https://www.ocg.at/de/wizik>, 23.04.2018

³⁰ <https://coderdojo.com/>, 23.04.2018

Möglichkeit zur Vermittlung von Programmierfertigkeiten. Eine wissenschaftlich fundierte Analyse, sowie eine Darlegung von Lernergebnissen, kompetenzbasierten Aktivitäten oder eine begründete und nachvollziehbare mögliche Förderung von Kompetenzen wurden nach eingehender Literaturrecherche von der Verfasserin der gegenständlichen Arbeit nicht gefunden.

Kodable stellt ein Curriculum für die eigene Programmiersprache für Kinder von der Vorschule bis zum Ende der Volksschule angepasst an den nordamerikanischen Lehrplan vor. Mittels einer iPad App und zahlreichen Unterrichtsmaterialien können Kindern, vorrangig der Volksschule, die Grundlagen der Programmierung vermittelt werden.³¹ Kodable ist kostenpflichtig, daher nicht frei verfügbar. Als Ergänzung kann für Schülerinnen und Schüler ab 11 bis 18 Jahren mittels dem Projekt CodeHS³² die Vermittlung von Programmiergrundlagen unterstützt werden. Bei Kodable werden die zu erreichenden Lernergebnisse angeführt, aber das Curriculum ist sehr stark auf Programmierung und weniger auf andere Aspekte der Informatik fokussiert. Die Zielgruppe ist mit 5 bis 10 Jahren zudem sehr unterschiedlich zu der der gegenständlichen Arbeit (3- bis 6-jährige Kinder). Es eignet sich somit nur bedingt zur frühen Technikbildung in Informatik für Kindergartenkinder.

Pekárová (2008) entwickelte ein Kursdesign von vier Einheiten zu je 20 bis 60 Minuten, basierend auf „Story-Telling“-Aktivitäten mit dem Bee-Bot für Vorschulkinder (5 bis 6 Jahre). Neben der Nutzung des Bee-Bots basteln die Kinder in den Einheiten auch und erstellen Materialien für die „Story-Telling“-Aktivitäten. (Pekárová, 2008, S. 120) In ihrer Forschung fand Pekárová (2008) heraus, dass Vorschulkinder kaum Berührungsängste beim Einsatz neuer Technologien zeigen, jedoch schnell die Motivation und das Interesse daran verlieren, wenn nicht konkrete Aufgabenstellung zur Nutzung motivieren. Darüber hinaus war die Gruppengröße pro Bee-Bot zu groß, so dass jedes Kind nur einmal einen Bee-Bot pro Einheit bedienen und nutzen konnte, während die anderen daneben saßen und zusahen. So kamen bei den anderen Kindern, die den Bee-Bot im Moment nicht nutzten, bald Ungeduld und Unaufmerksamkeit auf. (Pekárová, 2008, S. 117) Auch wenn Pekárová (2008) die durchgeführten Aktivitäten, im Gegensatz zu vielen anderen wissenschaftlichen Publikationen, näher erläutert, fehlt eine kompetenzbasierte empirische Evaluierung der Aktivitäten im Kontext von Kindergärten und eine Betrachtung des Beitrags der Aktivitäten zur Ausprägung des „Computational Thinkings“ bei den Kindern.

Für Volksschulkinder in der ersten Schulstufe (6- bis 7-jährige Kinder) gestalteten Cacco & Moro (2014) ein Lernprojekt, bei dem basierend auf der Geschichte, dass eine Biene zu einer Sonnenblume fliegen möchte, den Kindern Robotik und Programmierung nähergebracht werden. Zu Beginn wird ein Prototyp eines Roboters in Form einer Biene entwickelt, anschließend werden auf dem Papier mögliche Programmierungen erarbeitet. Abschließend wird anhand des realen Bee-Bots die Programmierung des selbigen erörtert und geübt. Cacco & Moro (2014) legen ausführlich die zu erreichenden Lernziele dar und beschreiben das Projektszenario, jedoch fehlt eine kompetenzbasierte Darstellung und weiterführende Projektideen für die Praxis, da das Thema (eine Biene möchte zur Sonnenblume fliegen) schnell erschöpft ist.

6.2 Research Gap: Computational Thinking und Educational Robotics

Es gilt die Lücke an einem kompetenzbasierten Framework zur Ableitung von Aufgaben für die Themengebiete „Educational Robotics“ und „Computational Thinking“ im Kindergarten zu schließen. Die Verfasserin der gegenständlichen Arbeit hat, neben den bisher angeführten Publikationen, weitere Arbeiten, die sich mit den Themen „Robotics in Education“,

³¹ <https://www.kodable.com/>, 23.04.2018

³² <https://codehs.com/>, 23.04.2018

„Educational Robotics“, „Computational Thinking“ (jeweils mit Fokus auf Kindergartenkinder) beschäftigt, gesucht und analysiert. Untersuchungskriterien zur Beurteilung der Publikationen auf ihre Relevanz hin waren:

- Die jeweilige Forschungsarbeit behandelt als Zielgruppe Kindergartenkinder im Alter von 3 bis 6 Jahren
- In der jeweiligen Forschungsarbeit werden konkrete Aufgabenstellungen oder Beschreibungen von Aktivitäten, die im Kindergartenalltag eingesetzt werden können, vorgestellt.
- Die in den jeweiligen Forschungsarbeiten vorgestellten Aufgaben verknüpfen Aufgaben ohne technische Hilfsmittel, Aufgaben mit Hardware-Unterstützung und Aufgaben mit Software-Unterstützung.
- Die Eignung der Aufgabenstellungen wurde im Rahmen der jeweiligen Forschungsarbeiten empirisch überprüft und belegt.
- In den jeweiligen Forschungsarbeiten wurden die Lernergebnisse dargestellt und die vorgestellten Aufgabenstellungen helfen dabei, diese Lernergebnisse zu erreichen.
- Neben den vorgestellten Aufgaben und den zu erzielenden Lernergebnissen werden in den Forschungsarbeiten auch die zu erwerbenden Kompetenzen erörtert.

Beraza, Pina & Demo (2010) zeigen weiterführende Einsatzmöglichkeiten von Bee-Bots im Kindergartenalltag, wobei die vorgestellten Aktivitäten eher dem Computational Thinking zuzuordnen sind als bei Pekárová (2008). Es ist zu hinterfragen, ob wirklich 4 bis 5-jährige Kinder bereits all diese Aktivitäten lösen können – es werden Mathematikkenntnisse benötigt, um zweistellige Zahlen zu addieren oder gar zu multiplizieren. Es werden in der Arbeit von Beraza, Pina & Demo (2010) erzielbare Kompetenzen vorgestellt, aber nur zwei exemplarische Aktivitäten näher angeführt. Eine empirische Evaluierung der erzielten Lernergebnisse ist nicht enthalten. Im Anhang ihrer Publikation räumen Beraza, Pina & Demo (2010) ein, dass es sich bei den teilnehmenden Kindern doch um Schülerinnen und Schüler einer zweiten Schulstufe handelt.

Gordon, Rivera, Ackermann & Breazeal (2015) legen ihre Arbeit an einem Roboter-Werkzeug-Set, das in einem sozialen, zwischenmenschlichen Kontext genutzt und in Kindergärten eingesetzt werden kann dar. Ziel ist es, über soziale Beziehungen Grundzüge des „Computational Thinkings“ zu vermitteln. Dazu werden vor allem für Kindergartenkinder vertraute Materialien zum Einsatz gebracht. (Gordon et. al., 2015, S. 1 f.) In ihrer Publikation wird aber nicht beschrieben, wie dieses Roboter-Werkzeug-Set konkret in der Praxis eingesetzt werden kann, welche Aufgaben damit bearbeitet werden können und welche Materialien tatsächlich zur Umsetzung benötigt werden.

Smith (2017) erforschte und gestaltete Hands-On Materialien für Kindergärten, mit denen Computational Thinking unter Einsatz der Montessori-Methode von den Kindern erlebt werden kann. In der Masterthesis von Smith (2017) am MIT wird keine Software-Unterstützung erwähnt. Im vorgestellten Curriculum im Anhang der Arbeit finden sich praktische Beispiele, die direkt in Lehr-Lern-Szenarien eingesetzt werden können, jedoch sind diesen keine Lernziele oder zu fördernde Kompetenzen zugeordnet.

Ein konzeptuelles Framework zur Gestaltung von Lehrszenarien unter Einsatz von Robotik und Programmierung stellen Misirli & Komis (2014) vor. Ihr Hauptkritikpunkt an den bisherigen Forschungstätigkeiten und -arbeiten ist, dass zwar Aktivitäten beschrieben werden, diese aber nicht in eine systematische Gestaltung der Lehre für die frühkindliche Entwicklung eingebettet sind. Daraufhin präsentieren Misirli & Komis (2014, S. 108) ein 7-Phasen Framework zur systematischen Gestaltung und Integration der Robotik- und Programmieraktivitäten in den Lehr-Alltag, jedoch sind diese Phasen allgemeiner, abstrakter Natur und daraus lassen sich weder Aufgabenstellungen für einen unmittelbaren Einsatz in der Praxis ableiten, noch erkennen, welche Materialien notwendig sind, um effektiv mit Kindergartenkindern an den Themen Robotik und Programmierung arbeiten zu können.

Weiter ist anzumerken, dass das 7-Phasen Framework für jegliche Art von Aus- und Weiterbildung in allen Altersstufen genutzt werden kann und dieses unabhängig von der jeweiligen Fachdisziplin ist. Es beinhaltet jedoch keinen Schwerpunkt explizit für Kindergärten und ist somit zu allgemein gehalten.

Hoffmann, Wendlandt & Wendlandt (2017) beschäftigen sich mit der Thematik, ob Kinder in der Grundschule (Volksschule) bereits grundlegend „algorithmisch Denken“ können. Sie fanden heraus, dass ein deutlicher Leistungsunterschied zwischen der zweiten und dritten Klasse besteht. Dieser zeigt sich vor allem in der Umsetzung des Konzepts „Iteration“. Leider werden kaum weiterführenden Aussagen zur ersten Schulstufe getätigt, die dem Kindergarten unmittelbar folgt, sowie keine Aufgaben, Aktivitäten oder Handlungsanweisungen zur Verfügung gestellt.

Bergner, Köster, Magenheim, Müller, Romeike, Schroeder & Schulte (2017) definieren Kompetenzbereiche zur frühen informatischen Bildung und konzentrieren sich nicht nur auf Kindergärten, sondern auch auf die Grundschule (Volksschule). Ihre Arbeit brachte eine Erweiterung der GI-Bildungsstandards (Gesellschaft für Informatik e.V.) um einen weiteren Prozessbereich hervor. Aus dieser Abhandlung sind jedoch keine konkreten Aufgaben ableitbar noch ist festzustellen, welche Hard- und Software zur Erreichung der gewünschten Kompetenzen zum Einsatz kommen kann.

Geldreich, Funke & Hubwieser (2017) stellen mit dem „Programmierzirkus“ einen speziellen Programmierkurs für Grundschulen (Volksschulen) vor, jedoch nicht für Kindergartenkinder. Der Kurs umfasst drei Tage zu je vier Stunden in den Klassen, wo Schülerinnen und Schüler erste Programmiererfahrungen sammeln können. Auch in dieser Publikation finden sich keine Aufgaben oder Handlungsanweisungen, woran konkret mit den Kindern im Unterricht gearbeitet werden soll.

Einen Überblick über die Möglichkeiten „Computational Thinking“ durch Programmierung zu lehren und zu lernen, liefern Lye & Koh (2014). Sie betrachten hierbei die Altersspanne von 6- bis 18-jährigen Kindern, keine Kindergartenkinder, und stellen keine Aufgaben oder Frameworks mit Aufgaben vor. Eine systematische Literaturrecherche über aktuelle Forschungstrends zum Einsatz von „Educational Robotics“ für junge Kinder (vom Kindergarten bis zum Ende der Volksschule) führten Jung & Won (2018) durch. Dabei konzentrierten sie sich in weiterer Folge ausschließlich auf den Einsatz von Robotern und Roboter-Sets im Lehr-Lern-Kontext. Sie zeigten zwar auf, dass es international noch wenig Forschung in Bezug auf die Zielgruppe Kindergartenkinder gibt (Jung & Won, 2018, S. 2), jedoch liefern sie ebenso, wie viele andere, keine Hinweise oder Handlungsanweisungen, wie „Educational Robotics“ nutzbringend in den Unterrichtsalltag eingebettet werden kann.

Fessakis, Gouli & Mavroudi (2013) zeigen in einer Case-Study auf, wie „Problem Solving“ im Kindergarten eingesetzt werden kann, damit Kinder Programmieren lernen können. Dazu wurden alle Kinder gleichzeitig unterrichtet und nur ein bestimmtes Software-Produkt eingesetzt. Somit konnte jedes der zehn teilnehmenden Kinder nur in jeweils einer Aktivität und nie mehrere Kinder gleichzeitig interagieren. Es zeigte sich, dass in diesem Setting die Kinder erhebliche Schwierigkeiten hatten, die gewünschten Programmierungen vorzunehmen. Die sieben durchgeführten Aktivitäten sind zwar rudimentär beschrieben, aber die dargestellten Erkenntnisse überzeugen nicht zu einem Praxiseinsatz selbiger. Darüber hinaus sind keine zu erwerbenden Kompetenzen erläutert oder konkrete Lernergebnisse formuliert.

Nach Hiller (2012) bestehen zwei Dimensionen in der Technikbildung: „eine ökonomische zur Talentförderung und zur erhöhten Akzeptanz der akademischen technischen Berufe (Rekrutierungsfunktion) und eine politische zur Förderung des technischen Interesses und technischer Basiskompetenzen, möglichst bereits in der Früherziehung.“ (Hiller, 2012, S. 157)

Sie arbeitet heraus, dass Kinder im Kindergarten und Volksschulbereich sehr interessiert an Naturwissenschaften und Technik sind und sich hier auch als Forscherinnen und Forscher verstehen. Jedoch können die Kinder ohne entsprechende Anleitung selten länger als 15 Minuten bei einer Aktivität zu einem Thema verharren. (Hiller, 2012, S. 167) Leider liefert Hiller (2012) in ihrer Publikation keine weiteren Hinweise dazu, wie Aktivitäten gestaltet werden sollten und welche Aktivitäten überhaupt möglich sind, um das Forschungsinteresse der jungen Kinder im Kontext der frühen Technikbildung in Informatik zu nutzen.

Eine provisorische Klassifizierung nach Nutzungsszenarien und Technologien für Roboter im Kontext von „Educational Robotics“ nehmen Catlin, Kandlhofer & Holmquist (2018) vor. Diese Klassifizierung ist jedoch noch in Entwicklung und liefert in ihrer derzeitigen Form noch zu wenig Anhaltspunkte für eine qualitativ hochwertige Auswahl von Robotern für Lehr-Lern-Szenarien. Sie ist daher im Moment für eher unerfahrene Lehrkräfte im Bereich „Computational Thinking“ und „Educational Robotics“ nicht zu empfehlen.

Cejka, Rogers & Portsmore (2006) berichten über den Einsatz von Robotern, im Speziellen Lego® RCX und der Software Robolab, im Kindergarten, betrachten aber dann recht zügig auch die Schulstufen 1 bis 5 näher und umreißen immer wieder die Altersspanne K-12, die vom Kindergarten bis zur 12. Schulstufe (in den USA) reicht. Eine Fokussierung auf Kinder im Alter von 3 bis 6 Jahren ist dieser Publikation nicht zu entnehmen. Ebenso werden die behandelten Themen vom Kindergarten bis zur 5. Schulstufe nur oberflächlich erwähnt, ohne darzustellen, welchen Aufgaben tatsächlich mit den Kindern und Schülerinnen und Schülern durchgeführt wurden.

Bers (2010) stellt mit dem „TangibleK Robotics Program“ ein Curriculum vor, das im Kindergarten, in Sommercamps und Laborumgebungen erprobt wurde. Das Curriculum besteht aus sechs „powerful ideas“, die in sechs Einheiten gelehrt werden sollen. Dazu stellt Bers (2010) auch eine Übersicht über die zu erreichenden Lernergebnisse und Kompetenzen zur Verfügung. (Bers, 2010, S. 10 f.) In weiterer Folge stellt sie einen Konnex zu anderen Fachdisziplinen her, die durch den Einsatz des Curriculums gelehrt werden. Jedoch fehlt in dieser Darstellung eine Auflistung konkreter Aufgaben und Tätigkeiten, um dieses Curriculum einfach und effektiv in die Praxis übernehmen zu können. Bers & Horn (2010) legen die Evaluierung einer tangiblen Programmiersprache und ein entsprechendes Curriculum für Kindergartenkinder dar. Sie erörtern 10 Einheiten näher, verzichten aber auf die Darstellung konkreter Aufgaben und Kompetenzen, die erworben werden können.

Kazakoff, Sullivan & Bers (2013) berichten, dass Kindern, im Speziellen hier jungen Volksschulkindern, durch Teilnahme an einem Kurs, der innerhalb einer Woche mit mind. 10 Unterrichtsstunden abgehalten wird, ermöglicht wird, das Bilden von Sequenzen in der Programmierung erlernen zu können. Die Autorinnen stellen zwar das übergeordnete Thema der Lehr-Lern-Einheiten vor, jedoch keine konkreten Aufgaben, wie das Lehr-Lern-Ziel erreicht werden kann. Siehe dazu auch Sullivan, Kazakoff & Bers (2013) oder Bers et al. (2014), wo zusätzlich belegt wird, dass Kinder in der Lage sind, unter Einsatz geeigneter Materialien einfache Programmierungen vorzunehmen. Hier zeigt sich, dass das Interesse und die Motivation der Kinder für die Thematik entsprechend hoch sind.

Elkin, Sullivan & Bers (2014) beschreiben ein ähnliches Projekt, wobei diesmal etwas ältere Kinder (7 bis 9 Jahre), in einem gemischten Setting nach Montessori an den Aktivitäten teilnahmen. Die Einheiten werden ausführlicher erörtert, jedoch nicht derart strukturiert, dass diese direkt in die Praxis übernommen werden können.

Sullivan & Bers (2015) listen mögliche Lernergebnisse, die nach einem 8-wöchigen „Robotics Curriculum“ erzielt werden können, auf. Hierzu berichten sie von 4- bis 7-jährigen Kindern, die an diesem Kurs teilgenommen haben: je eine Gruppe Kinder aus dem „Pre-Kindergarten“, dem Kindergarten, der ersten Schulstufe und der zweiten Schulstufe. In der Publikation werden die sieben Einheiten und das abschließende Projekt oberflächlich beschrieben sowie die eingesetzten Materialien ausführlich vorgestellt. Jedoch ist für Außenstehende nicht nachvollziehbar, was genau in den Lehr-Lern-Einheiten gemacht sowie

welche Aufgaben und Aktivitäten durchgeführt wurden, um die gewünschten Lernergebnisse zu erreichen.

Brennan & Resnick (2012) präsentieren die Schlüsseldimensionen des „Computational Thinkings“ („Computational concepts“, „Computational practices“, „Computational perspectives“) und wie die Erreichung dieser Dimensionen durch Lernende gemessen werden kann. Der Fokus liegt hierbei auf Kindern ab 8 Jahren und nicht auf Kindergartenkindern. Es werden in den Schlussfolgerungen der Arbeit zwar Links zu möglichen Lehr-Lern-Szenarien und Lernmaterialien genannt, diese beziehen sich aber ausschließlich auf Scratch als Werkzeug und auf die Zielgruppe ab Volksschule oder älter.

Eck, Hirschmugl-Gaisch, Hofmann, Kandlhofer, Rubenzer & Steinbauer (2013) erörtern einen sechs Einheiten umfassenden Kurs, der in Kindergärten zum Thema Robotik durchgeführt werden kann. Die Einheiten dauerten jeweils eine Stunde und Bee-Bots wurden als Lernmaterial eingesetzt. Eine weitere Aktivität ist ein sog. „Cross-generational robotics day in kindergarten“, der in einem anderen Kindergarten durchgeführt wurde. An diesem Tag nahmen Kindergartenkinder, Schülerinnen und Schüler (im Alter zwischen 11 und 13 Jahren) sowie Seniorinnen und Senioren teil. Hierbei konnten Kinder unterschiedliche Stationen besuchen und mehr über Robotik erfahren. In beiden Fällen finden sich nur oberflächlich beschriebene Themen in der Publikation, aber keine konkreten Aufgaben oder Handlungsanweisungen, wie die Themen vermittelt werden können. In Eck, Hirschmugl-Gaisch, Kandlhofer & Steinbauer (2014) ist eine inhaltliche Konzentration auf den „Cross-generational robotics day“ zu finden, wobei hier Bee-Bots in einer der zahlreichen Experimentier-Stationen zum Einsatz kamen. Konkrete Aufgabenstellungen, die in den Kindergartenalltag übernommen werden könnten, wurden aber auch hier nicht beschrieben.

Kandlhofer, Steinbauer, Hirschmugl-Gaisch & Huber (2016) beschreiben, wie mit Hilfe von vier Modulen die Themen Artificial Intelligence und Informatik durchgängig vom Kindergarten bis zur Universität gelehrt werden könnten. Pilot-Durchführungen sollen zeigen, dass dies im Kontext Österreich möglich ist. Im Kindergarten wurden hierzu 10 Einheiten zum Thema künstliche Intelligenz abgehalten. Die Lehrenden waren hierbei Schülerinnen und Schüler der Schule für Kindergartenpädagogik, die zuerst Lehreinheiten an der Universität zu dem Thema besuchten. Details zu den ausgeführten Einheiten bzw. eine Darstellung der durchgeführten oder abgehaltenen Aufgaben und Aktivitäten sind nicht publiziert.

Der Einsatz von Story-Telling im Educational Robotics ist zu empfehlen, so berichten Angel-Fernandez & Vincze (2018) über den Einsatz von Story-Telling als kreativitätsfördernde Maßnahme im Kontext von „Educational Robotics“ mit 6- bis 18-jährigen Kindern in Österreich. Sie belegen, dass Kinder ein sehr starkes Eigeninteresse an Robotik und Programmierung haben, auch außerhalb eines schulischen Kontexts. Sie betrachten in ihrer Forschung aber keine Kindergartenkinder und zeigen keine konkreten Aufgaben oder Aktivitäten.

Barr & Stephenson (2011) brechen eine Lanze für die durchgängige Integration von „Computational Thinking“ in die Lehre, vom Kindergarten bis zur zwölften Schulstufe. Sie zeigen zudem auf, welche zahlreichen multidisziplinären Anknüpfungspunkte „Computational Thinking“ mit sich bringt, gehen aber in keiner Weise darauf ein, wie, gerade im Kindergarten, das „Computational Thinking“ integriert und gelehrt werden kann. Grover & Pea (2013) berichten ähnliches. Sie listen zwar vier allgemein verfügbare Frameworks bzw. Curricula auf, die in Schulen eingesetzt werden können, jedoch ist die Zielgruppe K-12 (Kinder im Alter von 6 bis 18 Jahren, in den USA) nicht die Zielgruppe der gegenständlichen Arbeit. Daher ist die Eignung der erwähnten Frameworks oder Curricula für Kindergartenkinder im Alter von 3 bis 6 Jahren zu hinterfragen, da Lese- und Schreibkompetenzen bei den teilnehmenden Kindern vorausgesetzt werden.

Auch Mannila, Dagiene, Demo, Grgurina, Mirolo, Rolandsson & Settle (2014) listen drei (zum Teil die gleichen wie oben) Frameworks bzw. Curricula auf. Das Angebot von code.org beinhaltet hierzu auch Aufgaben und Übungen für Kindergartenkinder, jedoch bedarf es hier

einer Betreuungsperson, die die jeweiligen Aufgabenstellungen vorlesen kann. Außerdem benötigen die Kinder zur Lösung der meisten Aufgaben einen PC mit Internetverbindung. Das erwähnte CSTA Framework (siehe auch Kapitel 3.1.2) erfordert von den Lehrkräften einiges an Informatik- und „Computational Thinking“- Hintergrundwissen, um die darin enthaltenen Aktivitäten im Kindergarten einsetzen zu können. Darüber hinaus zielen alle genannten Frameworks oder Curricula darauf ab, eine Durchgängigkeit im K-12 System (also Kinder im Alter von 6 bis 18 Jahren) zu ermöglichen und bieten daher keine fokussierte Sicht auf Kindergartenkinder an.

Tsukamoto, Takemura, Nagumo, Ikeda, Monden & Matsumoto (2015) stellen ein Projekt vor, mit dem Volksschulkinder unter Einsatz einer textbasierten Programmiersprache Programmieren lernen sollen. Dazu standen zwei Lehr-Lern Einheiten mit einer Dauer von je einer Stunde zur Verfügung. Ob Kinder nun wirklich damit Programmieren lernen können, bleibt offen, denn die Autorinnen und Autoren untersuchten hauptsächlich die Motivation der beteiligten Schülerinnen und Schüler sowie die Fähigkeiten der Kinder, Koordinaten und Pixel zu verstehen.

Kucuk & Sisman (2017) untersuchten das Verhalten von Volksschulkindern (im Alter von 8 bis 11 Jahren) in 1:1 Lehr-Lern-Szenarien zum Thema Robotik. Sie liefern mit ihren Ergebnissen wertvolle Hinweise zur Gestaltung von Rahmenbedingungen, jedoch keine Übersicht über die eingesetzten Aufgaben oder welche konkreten Aktivitäten in den jeweiligen Lehr-Lern-Szenarien untersucht wurden. Darüber hinaus beschäftigten sie sich nicht mit Kindergartenkindern.

Wie „Computational Thinking“ gelehrt und gelernt werden kann, beantworten Hsu, Chang & Hung (2018) mit einer umfassenden Literaturrecherche, deren Ergebnis eine Auflistung von 16 Lernstrategien im Kontext von „Computational Thinking“ darstellt. Ergänzt wird diese durch eine Präsentation der eingesetzten Programmiersprachen. Die Autorinnen und Autoren betrachten „Computational Thinking“ als eine wichtige Kompetenz, die auch einen wesentlichen Beitrag beim Erwerb anderer Fachkompetenzen leisten kann. Es werden jedoch keine Lehr-Lern-Materialien, Lernergebnisse oder Aktivitäten vorgestellt, die einen direkten Einsatz von „Computational Thinking“ im Lehr-Alltag ermöglichen.

Gomes, Falcão & Tedesco (2018) untersuchten in einer brasilianischen Schule im Informatikunterricht mit 5- bis 7-jährigen Kindern, wie diesen mit Hilfe von digitalen Spielen Programmierkonzepte vermittelt werden können. Auch wenn die Zielgruppe ähnlich zu der, der gegenständlichen Forschungsarbeit ist, sind die eingesetzten Materialien und die Rahmenbedingungen different. Bei Gomes, Falcão & Tedesco (2018) wird Informatikunterricht für 5- bis 7-jährige Kinder in der Schule angeboten, etwas, was man in unseren Breitengraden noch sehr selten vorfindet. Und zum anderen werden Spiele zur Vermittlung der Inhalte gewählt und nicht wie in der gegenständlichen Arbeit eine Kombination aus einem Bodenroboter und einer Software-Simulation desselben. In der Publikation werden die Lehrinhalte oberflächlich beschrieben und die eingesetzten digitalen Spiele bewertet, jedoch werden keine konkreten zu erzielenden Lernergebnisse oder konkrete Kompetenzen, die gefördert werden sollen, angeführt bzw. überprüft.

Ein mögliches Framework zur Integration von „Computational Thinking“ in Curricula stellen Angeli, Voogt, Fluck, Webb, Cox, Malyn-Smith & Zagami (2016) vor. Sie konzentrieren sich hierbei auf drei Alterskategorien (6 bis 8 Jahre, 9 bis 10 Jahre, 11 bis 12 Jahre). Im Framework werden die zu lehrenden Themen mit abstrakten Inhaltsbeschreibungen diesen Alterskategorien zugeordnet. Es konzentriert sich nicht ausschließlich auf Kindergartenkinder und zeigt keine konkreten Aufgabenstellungen oder Beschreibungen von Aktivitäten, die mit den betreffenden Kindern genutzt werden können. Es wird jedoch aufgezeigt, welche Kompetenzen Lehrkräfte benötigen, um „Computational Thinking“ zu lehren. Die Autorinnen und Autoren haben in Folge ein Ausbildungsprogramm für Lehrkräfte, basierend auf ihrem Curriculum entwickelt, aber geben kaum Auskunft darüber, welche konkreten Inhalte hierbei vermittelt werden.

Döbeli Honegger & Hielscher (2017) zeigen die Herausforderungen auf, denen Lehrpersonen bei der Integration von neuen Lehrinhalten, wie z.B. „Computational Thinking“, gegenüberstehen. „Auf Grund der fehlenden eigenen Erfahrungen braucht es konkrete Beispiele, um sowohl eine Vorstellung zu entwickeln, wie Informatik in der Volksschule aussehen kann, als auch das Selbstvertrauen der Lehrpersonen zu stärken, diese Inhalte selbst umsetzen zu können.“ (Döbeli Honegger & Hielscher, 2017, S. 106) Dies lässt sich ebenso auf den Kindergarten als Bildungsinstitution übertragen. Ähnliches berichten Israel, Pearson, Tapia, Wherfel & Reese (2015).

Zusammengefasst kann festgehalten werden, dass sich Publikationen entweder mit Kindergärten und Volksschulen oder gar höheren Schulen beschäftigen, folglich nicht ausschließlich die Zielgruppe Kindergärten bzw. Kinder im Alter von 3 bis 6 Jahren bedienen. Oder es werden keine konkreten Aufgaben bzw. Handlungsanweisungen gegeben, wie Lehr-Lern-Szenarien zur Vermittlung von „Computational Thinking“ aussehen und welche Aufgaben und Aktivitäten eingesetzt werden können. Den meisten Publikationen sind beide Aspekte gemein. Leider ist somit schwer nachvollziehbar, welches Wissen tatsächlich den jeweiligen Kindern vermittelt wurde und welche Lernergebnisse dabei verfolgt wurden. Weiters zeigt kaum eine Publikation auf, welche Kompetenzen tatsächlich durch die Lehre des „Computational Thinkings“ gefördert werden können.

Die gegenständliche Forschungsarbeit leistet einen Beitrag zur Schließung der dargestellten Lücke in der wissenschaftlichen publizierten Forschungslandschaft mit der Präsentation eines empirisch erforschten, kompetenzbasierten Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ mit Aktivitätsschablonen für Kindergärten zur frühen Technikbildung in Informatik. Mit Hilfe dieses Frameworks soll es nicht nur Kindergartenpädagoginnen und -pädagogen, sondern auch zum Kindergarten Dritten, die in diesem Themengebiet arbeiten möchten, ermöglicht werden, auf einfache Art und Weise Aufgaben aus erprobten und empirisch befundeten Aktivitätsschablonen zu generieren und diese regelmäßig im Kindergartenalltag einzusetzen. So kann eine wertvolle Mitwirkung zur frühen Technikbildung in Informatik ermöglicht werden. Mehr dazu ist Kapitel 8 zu entnehmen.

6.3 Research Gap: Child-Centered-Design zur frühen Technikbildung in Informatik

Im Bereich des Child-Centered-Designs, welches zur frühen Technikbildung in Informatik eingesetzt werden soll, ist eine Forschungslücke zu schließen, wie die folgende Darstellung zeigt. Gestützt wird die Definition der Forschungslücke durch folgende Untersuchungskriterien in Korrelation zu den Forschungsfragen (siehe Kapitel 1.3.2.1), insbesondere F2.4, zur Beurteilung der Publikationen auf ihre Relevanz:

- Die jeweilige Forschungsarbeit behandelt als Zielgruppe Kindergartenkinder im Alter von 3 bis 6 Jahren.
- In der jeweiligen Forschungsarbeit wird das Lernen durch den Child-Centered-Design-Prozess in den Vordergrund gestellt.
- Das Lernen durch Teilnahme am Child-Centered-Design-Prozess dient der frühen Technikbildung und erfolgt kompetenzbasiert bzw. werden Lernergebnisse und mögliche zu fördernde Kompetenzen dargelegt.

Wie in Kapitel 5.4 bereits erwähnt, wirkten Kinder im Alter von 4 bis 6 Jahren bei der Entwicklung des sog. „Sneetches Room“ mit. (Montemayor et al., 2002) Lernen durch die Teilnahme am Gestaltungsprozess bzw. das Erreichen von Lernzielen wird in der Arbeit von Montemayor et al. (2002) jedoch nicht behandelt. Auch bei Cairncross & Waugh (2005), die mit 3- bis 4-jährigen Kindern arbeiteten, steht die Teilnahme der Kinder an der Gestaltung einer Lernapplikation im Vordergrund und Lernen durch eine Teilnahme am Gestaltungsprozess bzw. die Förderung von Kompetenzen durch eine Teilnahme am Gestaltungsprozess werden nicht betrachtet. Gleiches gilt für die Arbeit von Kindborg & McGee (2007).

Read & Markopoulos (2013) leiteten 2013 das „Journal of Child-Computer Interaction“ mit einem Editorial ein. Mit Child-Computer Interaction (CCI) wird hierbei die wissenschaftliche Betrachtung der Interaktion zwischen Kindern und Computern und Kommunikationstechnologien betrachtet. Die CCI ist eine multidisziplinäre Forschungsrichtung, die Kinder auch in unterschiedlichen Rollen bei der Nutzung von Informations- und Kommunikationstechnologien betrachtet. (Read & Markopoulos, 2013, S. 2) In diesen Forschungsbereich fällt auch die Betrachtung der Kinder und ihrer Partizipation in den Rollen als Gestalterinnen und Gestalter sowie als Nutzerinnen und Nutzer. (Read & Markopoulos, 2013, S. 4) Kinder werden hierbei jedoch nicht explizit als Lernende durch den Designprozess betrachtet.

Bekker, Bakker, Douma, van der Poel & Scheltenaar (2015) beschäftigen sich in ihrer Arbeit mit der Frage, wie Kindern digitale Kompetenzen durch Design-Based Learning mit digitalen Werkzeugsets vermittelt werden können. Sie betrachten hierzu Case Studies mit Schulkindern im Alter von 11 bis 14 Jahren, die nicht der Zielgruppe der gegenständlichen Forschungsarbeit entsprechen. Sie fokussieren sich unter anderem auf die Integration von Reflexion als wesentlichen Bestandteil dieses Lernprozesses. (Bekker et al., 2015, S. 30) Die Autorinnen und Autoren liefern mit ihrer Arbeit wertvolle Hinweise zur Gestaltung solcher Lehr-Lern-Szenarien, jedoch fehlt eine lernzielorientierte oder kompetenzbasierte Darstellung, was die Kinder durch die Teilnahme an Design-Based Learning mit digitalen Werkzeugkästen tatsächlich lernen.

Auch Giannakos & Jaccheri (2018) beschäftigen sich in ihrer Arbeit damit, was man durch Einsatz von kreativen Entwicklungsaktivitäten zur Gestaltung von Spielen lernen kann. Jedoch werden wiederum Kinder betrachtet, die mit 12 bis 17 Jahren außerhalb der Zielgruppe der gegenständlichen Arbeit liegen. Es zeigt sich aber bei Giannakos & Jaccheri (2018), dass der Einsatz solch kreativer Aktivitäten das Bewusstsein für technische Möglichkeiten bei den Teilnehmerinnen und Teilnehmern steigert. Zusätzlich werden die Grenzen der technischen Machbarkeiten ausgelotet, die Zusammenarbeit mit anderen Teilnehmerinnen und Teilnehmern intensiviert und unterschiedliche Ansichten und Erfahrungen ausgetauscht. (Giannakos & Jaccheri, 2018, S. 34) In der Arbeit werden zwar die eben genannten Erkenntnisse begründet dargelegt, aber keine Lernziele und zu fördernde Kompetenzen zugeordnet.

Schepers, Dreessen & Zaman (2018) betrachten 6- bis 10-jährige Kinder, nicht die Zielgruppe der gegenständlichen Arbeit, als Prozessdesigner des Gestaltungsprozesses von interaktiven Materialien und erweitern damit das klassische Participatory Design (siehe Kapitel 5.2.3.1). Sie fordern ein Aufbrechen der klassischen Herangehensweisen und ein Überdenken der Rollen der Kinder in solchen Gestaltungsprozessen, da sie davon überzeugt sind, dass Kinder noch intensiver partizipieren, wenn sie andere oder erweiterte Rollen einnehmen dürfen. (Schepers, Dreessen & Zaman, 2018, S. 53) Die Arbeit von Schepers, Dreessen & Zaman (2018) zeigt keine Lernergebnisse oder mögliche zu fördernde Kompetenzen auf. Auch die Rolle der Lernenden, die Kinder einnehmen könnten, wird nicht diskutiert.

Kinnula, Iivari, Isomursu & Kinnula (2018) verfolgen den Ansatz, dass Kinder, die an Design-Prozessen teilnehmen, auch einiges über Design bzw. Gestaltung erlernen. Sie analysieren mit dem Value Creation Ansatz (und der Role-based Value Creation Analysis-Methode) vier Projekte (eine Lernportfolio-Anwendung und drei Spiele) mit 8- bis 14-jährigen Kindern (was nicht der Zielgruppe der gegenständlichen Arbeit entspricht). Sie identifizieren in ihrer Forschung jede „wertvolle“ Rolle, die Kinder im Design-Prozess einnehmen und verallgemeinern diese Rollen auf universelle Werte sowie motivationale Bedürfnisse. (Kinnula et al., 2018, S. 43 ff.) Es zeigt sich, dass Kinder deutlich mehr Rollen einnehmen als bisher, z.B. in „Druin's Model“, angenommen. (Kinnula et al., 2018, S. 45) In der Forschungsarbeit von Kinnula et al. (2018) werden weder zu erreichende Lernziele noch zu fördernde Kompetenzen angeführt.

Landoni, Rubegni & Nicol (2018) unterstützen die Aussagen und Erkenntnisse von Schepers, Dreessen & Zaman (2018) und Kinnula et al. (2018). Landoni, Rubegni & Nicol (2018) berichten von ähnlichen Erkenntnissen aus zwei unterschiedlichen Projekten mit 8- bis 12-jährigen Kindern, was nicht der Zielgruppe der gegenständlichen Forschungsarbeit entspricht. Sie konzentrieren sich auf die Untersuchung der Motive, Motivationen und Bedingungen für die aktive Teilnahme der Kinder, und nicht auf mögliche zu erzielende Lernergebnisse oder einen möglichen Kompetenzerwerb oder die Förderung von Kompetenzen.

Barendregt, Bekker, Börjesson, Eriksson & Torgersson (2016) regen an, dass Design-Prozesse mit Kindern um Lernziele erweitert werden sollten um das Lernen in Design-Prozessen transparent und messbar zu machen. Sie führten mehrere Projekte mit Master-Studierenden durch, die gemeinsam mit Schulkindern an Design-Projekten arbeiteten, mit dem Ziel Technologie zur sozialen-emotionalen Entwicklung zu nutzen. Über die teilnehmenden Kinder gibt es keine Altersangaben in der Publikation, jedoch verdichtete Hinweise auf Schulkinder in frühen Schuljahren, welche nicht der Zielgruppe der gegenständlichen Arbeit entsprechen. Barendregt et al. (2016) fanden in ihrer Studie heraus, dass es zwar schwer ist entsprechende Lernziele zu finden, aber diese hilfreich sein können um Kinder zur Teilnahme an Design-Prozessen zu motivieren, wenn den Kindern die Lernziele kommuniziert und die Möglichkeiten zum Lernen und zur Reflektion während den Aktivitäten geboten werden. Konkrete Lernziele werden nicht vorgestellt und eine mögliche Förderung von Kompetenzen wird nicht behandelt.

Leider konnten keine Studien zur Usability-Evaluierung gängiger Bodenroboter oder Software Simulationen von Bodenrobotern durch die Zielgruppe selbst gefunden werden. Es gilt diese Lücke exemplarisch für den populären Bodenroboter Bee-Bot und die korrespondierende Software Simulation Focus on Bee-Bot 1 & 2 zu schließen.

Zum Ansatz, Child-Centered-Design als Methode zur Wissensvermittlung zu nutzen oder Lernen durch den Einsatz von Child-Centered-Design für 3- bis 6-jährige Kinder zu fördern, wurden nach intensiver Literaturrecherche von der Verfasserin der gegenständlichen Arbeit keine wissenschaftlichen Publikationen gefunden. Es ist daher an dieser Stelle von einer wesentlichen Neuigkeit und Innovation auszugehen, wenn in der gegenständlichen Arbeit Child-Centered-Design als Maßnahme zur frühen Technikbildung in Informatik eingesetzt wird. Dies stellt einen wesentlichen Beitrag zur Schließung der skizzierten Lücke in der wissenschaftlichen Publikationslandschaft dar. Mehr dazu ist Kapitel 10 zu entnehmen.

6.4 Zwischenfazit: Frühe Technikbildung in Informatik

Levy & Mioduser (2008) untersuchten, ob Kinder im Kindergartenalter von 5 bis 6 Jahren in der Lage sind, das Verhalten eines Roboters als Maschine wahrzunehmen und die Abhängigkeiten von Bedingungen und Handlungen zu erkennen. Interessant dabei ist, ob Kindergartenkinder die konzeptuellen Modelle von technologischen Systemen verstehen und artikulieren können. (Levy & Mioduser, 2008, S. 337 f.) Sie fanden heraus, dass Kinder ohne zusätzliche Hilfe in der Lage sind, einfache Ursache-Wirkungs-Beziehungen technisch zu formulieren. Werden die Zusammenhänge und Beziehungen zwischen den Bedingungen und Handlungen komplexer, dann wechseln Kinder von selbst zu psychologischen Erklärungen (das Verhalten des Roboters wird wie das eines Menschen, basierend auf Gefühlen u.ä. beschrieben). Dies geschieht nach Levy & Mioduser (2008), weil aus dieser Sichtweise das Verhalten des Roboters kürzer und prägnanter beschrieben werden kann. (Levy & Mioduser, 2008, S. 352)

Beran & Ramirez-Serrano (2010) zeigen, dass Kinder Roboter eher als menschenähnlich wahrnehmen, je autonomer die Roboter sind und je mehr diese komplexe Vorgänge bzw. Bedingungen für Handlungen berücksichtigen können. (Beran & Ramirez-Serrano, 2010, S. 138) Zunehmende Intervention durch Erwachsene oder Lehrkräfte hilft Kindern das Verständnis für die technischen Zusammenhänge zu entwickeln. Sie können das Handeln

von Robotern eher abstrahieren und die dahinterliegenden Konzepte artikulieren. (Levy & Mioduser, 2008, S. 354)

Wie die in diesen ersten Kapiteln der gegenständlichen Arbeit angeführten Beispiele und zahlreiche weitere Studien belegen, ist der korrekte und sinnbringende Einsatz von Technologien, insbesondere IKT, in elementaren Bildungseinrichtungen für Kinder von Vorteil und hilft ihnen im weiteren Verlauf (ihres Bildungsweges) selbstsicherer zu werden und leichter zu lernen. Siehe dazu u.a. Rice & Slaten (2010). Auch Crawford, Wood, Fowler & Norrell (1994) zeigen auf, dass Lernen nicht nur durch Konsumation von Inhalten stattfindet, sondern dass aktive Partizipation der Lernenden zur Motivation derselben beiträgt. „Students will understand math and sciences through the challenge of doing and applying, not by passively observing and listening from a seat in a classroom.“ (Crawford et al., 1994, S. 180)

Die Fähigkeit des Menschen, über „eine praktisch unendliche Menge von Objekten rational zu argumentieren, [...] wird [...] mit zunehmender Komplexität der technischen Infrastruktur unserer Gesellschaft immer wichtiger und sollte deshalb in der Ausbildung stärker gepflegt werden.“ (Nievergelt, 1999, S. 365) Jedoch sind (technik- und natur-) wissenschaftliche Aspekte im Unterricht, gerade zu Beginn der Schulpflicht und in frühen Klassen, unterrepräsentiert. (Streim, Horwitz, Pica & Abel, 1995, S. 1) Die Integration des Computers in den Kindergarten und die ersten Schuljahre leistet jedoch einen wesentlichen Beitrag zur Medienbildung und reduziert frühzeitig individuelle Defizite bei Schülerinnen und Schülern. (Herper & Hinz, 2009, S. 85)

Die Auseinandersetzung mit Wissenschaft und eine positive Lernerfahrung im Kindergartenalter haben in zahlreichen Langzeitstudien positive Effekte auf die Lernleistung und (Wissenschafts-)Begeisterung der Kinder im späteren Leben gezeigt. (Saçkes et al., 2010, S. 219) Auch Samarapungavan, Mantzicopoulos & Patrick (2008) belegen, dass Kindergartenkinder in der Lage sind, erste wissenschaftliche Zusammenhänge zu bilden und Argumentationen abzuleiten, wenn sie entsprechende Aktivitäten besuchen konnten. (Samarapungavan, Mantzicopoulos & Patrick, 2008, S. 900 f.)

Bers & Horn (2010) erachten es als notwendig, dass sich Kinder mit Technologie auseinandersetzen. Sie sollten nicht nur ein Verständnis für die natürliche Welt entwickeln, sondern sich auch mit technologischen Entwicklungen beschäftigen, da sie in ihrem gesamten Alltag davon umgeben sind. (Bers & Horn, 2010, S. 49) Bislang scheitert dies aber an für Kinder geeigneten Tools sowie Lernmaterialien und an der Tatsache, dass technologische Entwicklungen sehr komplexe und abstrakte Phänomene repräsentieren. (Bers & Horn, 2010, S. 50) Mit den richtigen Mitteln, den richtigen Aufgaben und im geeigneten Kontext ist es nach Bers & Horn (2010) möglich, Kindern im Alter von 5 Jahren und älter die Welt der Technik näher zu bringen. Daher stellt sich heute nicht mehr die Frage, ob Kindern IKT nähergebracht werden soll, sondern wie. (Bers & Horn, 2010, S. 50 ff.) Diese Aussage bzw. Erkenntnis tätigen Clements & Nastasi (1992) bereits Anfang der 1990er Jahre. Die wichtige Frage ist, wie Computer am Besten eingesetzt werden können, um die Entwicklung junger Kinder optimal zu unterstützen. (Clements & Nastasi, 1992, S. 233)

Da für die spätere Entwicklung der Kinder dem Kindergarten-, Vorschul- und Volksschulalter eine große Bedeutung zukommt, sollte ihr vom Bildungssystem große Aufmerksamkeit geschenkt werden. Die frühe Auseinandersetzung mit Naturwissenschaften, Technik und Mathematik wird auch im BildungsRahmenPlan für Österreich (Hartmann et al., 2009) hervorgehoben.

Die aus der Literatur abgeleitete Kritik am Bildungssystem umfasst vor allem die Kritik an der Gestaltung der Lehre, in der Lern- und Prüfungssituationen oft nicht klar getrennt sind. Dadurch werden die Lernenden zu einem „learning to the test“ gedrängt und eine wirkliche Kompetenzentwicklung behindert. Der Forderung von Prenzel, Reiss & Hasselhorn (2009) nach Freiräumen in der Ausbildung, in denen auch Fehler von Lernenden gemacht werden

dürfen, die zu ihrer Weiterentwicklung dienen, ist vollinhaltlich zuzustimmen. Diese Art zu Lernen ist gerade in elementaren Bildungseinrichtungen von den organisatorischen Rahmenbedingungen her sehr gut möglich. Die mangelnde Ausbildung der Pädagoginnen und Pädagogen im elementaren Bildungsbereich in Bezug auf technische und mathematische Disziplinen, die auch in der Literatur häufig kritisiert wird (siehe Kapitel 2.3), stellt allerdings eine Herausforderung dar.

Nach Jung (2010) lässt sich die Kompetenzentwicklung wie folgt beschreiben: Die durch Herausforderungen entstandenen Bedürfnisse, Motive und Ziele eines Individuums motivieren zu den eigenen Handlungen. Die dabei gewonnenen Erfahrungen fließen in den Aufbau von Fähigkeiten, Fertigkeiten und Kenntnissen mit ein. Entwickelte Normen, Werte und Einstellungen beeinflussen in weiterer Folge wiederum die eigenen Bedürfnisse, Motive und Ziele. (Jung, 2010, S. 27 f.)

Wie in den bisherigen Kapiteln der gegenständlichen Forschungsarbeit dargelegt, ist die Nutzung von IT und auch Robotern im frühen Kindesalter (3 bis 6 Jahre) nicht nur der Allgemeinbildung und der IT-Bildung dienlich, sondern fördert auch weitreichende Kompetenzen, die für den Alltag des Menschen unabdingbar sind.

7 Forschungsmethoden

Die Darstellung der gewählten Forschungsmethoden basiert vorwiegend auf der Literatur von Bortz & Döring (2006) sowie Mayring (2002 sowie 2010). In Kapitel 7.1 stellt die Verfasserin der gegenständlichen Arbeit die qualitative Feldforschung vor, die als Grundlage zur Erstellung des notwendigen Datenmaterials dient. Für die Analyse und Ergebnisfindung setzt die Autorin die qualitative Inhaltsanalyse nach Mayring ein (Kapitel 7.2), die es erlaubt qualitatives Untersuchungsmaterial möglichst objektiv und nachvollziehbar auszuwerten.

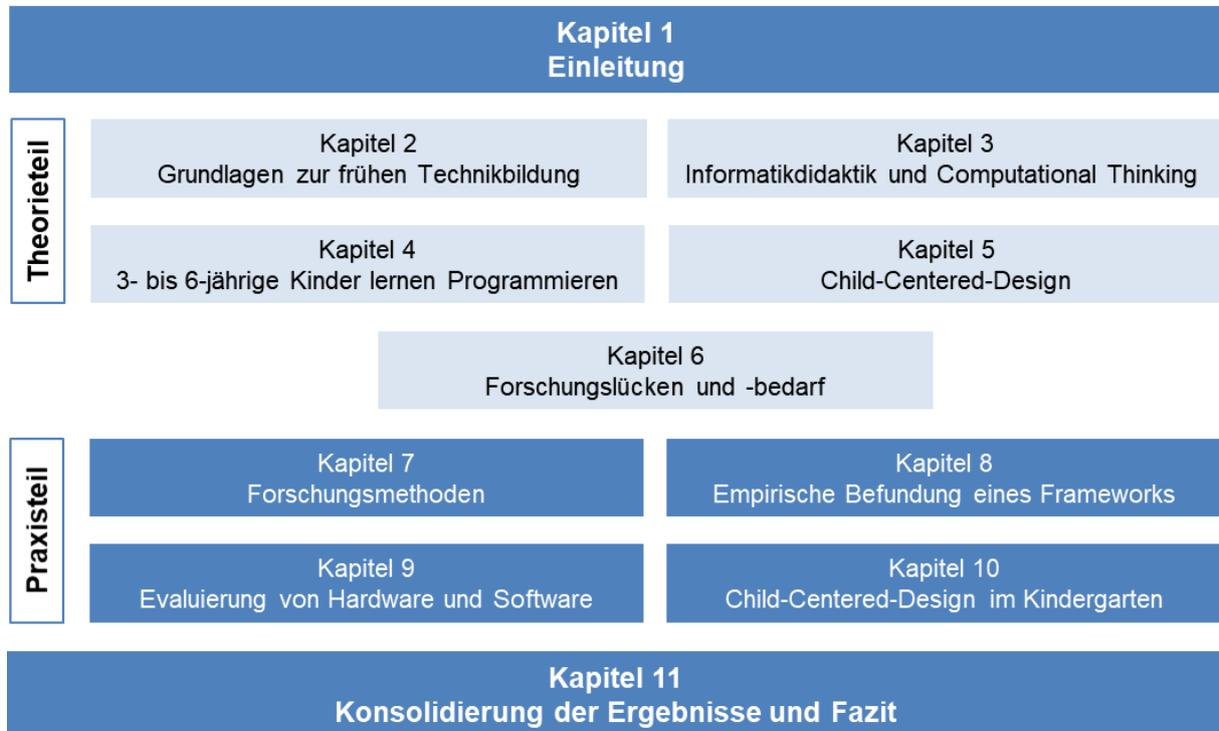


Abbildung 7: Kapitelstruktur Praxisteil

In Kapitel 8 behandelt die Verfasserin der gegenständlichen Arbeit die Entwicklung und empirische Befundung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Dazu legt die Autorin nicht nur den Prozess dar, dem sie folgte, sondern definiert auch Kompetenzen bzw. Lernergebnisse, die durch die Anwendung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ erworben bzw. gefördert werden sollen. Im Anschluss an die Präsentation der gewonnenen empirischen Erkenntnisse über das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ im Einsatz in Kindergärten zur frühen Technikbildung in Informatik wird das finale Framework vorgestellt.

Die Evaluierung der eingesetzten Hard- und Software ist in Kapitel 9 dokumentiert. Hierzu beschreibt die Verfasserin der gegenständlichen Arbeit zum einen die eingesetzten Methoden und das jeweilige Testsetting und zum anderen die Ergebnisse bzw. empirischen Erkenntnisse im Kontext der frühen Technikbildung in Informatik, bei der die Evaluierung der Hardware und Software eine Doppelfunktion einnimmt. Einerseits dient die Evaluierung der Usability der Hard- und Software zur Bewertung selbiger und andererseits als Methode bzw. Werkzeug zur frühen Technikbildung in Informatik, bei der Kinder erfahren und erleben können, dass Informatik mehr ist als „nur“ Programmieren.

Child-Centered-Design, als zweiten Praxisschwerpunkt der gegenständlichen Arbeit, behandelt die Autorin theoretisch in Kapitel 5 und der korrespondierende praktische Teil findet sich in Kapitel 10. Neben dem Ablauf des Child-Centered-Design-Prozesses werden die unterschiedlichen Detailaktivitäten wie die Gestaltung von Symbolen, Card-Sorting, Prototyping und die Evaluation des erstellten Prototyps erörtert. Zum Abschluss des Kapitels

führt die Verfasserin der gegenständlichen Arbeit eine Betrachtung des Beitrags eines adaptierten Child-Centered-Designs zur frühen Technikbildung in Informatik durch.

Zu guter Letzt konsolidiert die Autorin die empirischen Erkenntnisse aus dem Praxisteil mit den Erkenntnissen aus dem Theorieteil und legt eine Zusammenfassung der wichtigsten Ergebnisse vor (Kapitel 11).

7.1 Die qualitative Feldforschung

Die qualitative Feldforschung beschäftigt sich, im Gegensatz zur Untersuchung im Labor, mit dem Feld, also der natürlichen Umgebung des Beobachtungsgegenstandes. (Bortz & Döring, 2006, S. 336) Felduntersuchungen weisen im Gegensatz zu Laboruntersuchungen in der Regel eine hohe externe Validität auf. Dies führt jedoch gleichzeitig zu einer geringeren internen Validität, da störende Einflussgrößen nur bedingt bis gar nicht kontrolliert werden können. Es sind daher in der Praxis entsprechende Kompromisslösungen zu suchen, die sowohl die externe als auch die interne Validität zufriedenstellend berücksichtigen. Die experimentelle Felduntersuchung erscheint in Folge als sinnvolle Kombination. (Bortz & Döring, 2006, S. 57) „Bei experimentellen Untersuchungen werden Untersuchungsobjekte per Zufall in Gruppen eingeteilt (Randomisierung), bei quasiexperimentellen Untersuchungen arbeitet man mit natürlichen Gruppen.“ (Bortz & Döring, 2006, S. 54)



Abbildung 8: Phasen der qualitativen Feldforschung

Die Arbeitsschritte der qualitativen Feldforschung lassen sich, wie in Abbildung 8 dargestellt, in sechs wesentliche Phasen zusammenfassen:

1. Planung und Vorbereitung
Neben der organisatorischen Planung (Verfügbarkeiten, Zeit, Finanzierung, Probandinnen und Probanden, ...) ist hier auch der Fokus auf die inhaltliche Planung zu legen. (Bortz & Döring, 2006, S. 338)
2. Einstieg ins Feld
Das Feld wird nach seiner Zugänglichkeit in offene, halboffene und geschlossene Bereiche unterteilt. Dabei ist unter offener Bereich ein Bereich mit in der Regel großer Öffentlichkeit (z.B. öffentlicher Raum wie Straßen, Plätze, ...), unter halboffener ein Bereich mit einer Personenmenge einer Organisation (z.B. Universität, Schule) und unter geschlossener Bereich einer mit einer klar abgegrenzten Personenmenge (z.B. einer Schulklasse in einem Raum oder mit einem Gerichtssaal) zu verstehen. Zumeist müssen die Forscherinnen und Forscher in das Feld eingeführt werden: die Erlaubnis zur Beobachtung und Teilnahme im Feld muss eingeholt werden, den Probandinnen und Probanden werden die Forscherinnen und Forscher vorgestellt und das Ziel der Feldforschung präsentiert. (Bortz & Döring, 2006, S. 338)

3. Agieren im Feld

Während der Tätigkeit im Feld ist darauf zu achten, dass die Forscherin oder der Forscher selbstkritisch in ihrer, seiner Doppelfunktion als beobachtende und teilnehmende Person agiert. Je länger eine Beobachtung dauert, desto mehr wird die beobachtende Person von den übrigen Personen als „normal“ und „dazugehörend“ wahrgenommen. (Bortz & Döring, 2006, S. 339) Es ist jedoch Vorsicht geboten, dass es zu keinem Distanzverlust kommt und die Forscherin oder der Forscher in ihrer Wahrnehmung nicht beeinträchtigt wird. (Bortz & Döring, 2006, S. 340)

4. Dokumentation der Feldtätigkeit

Die Dokumentation der Feldtätigkeit kann alleine oder im Team erfolgen. Gezieltes Nachfragen und der Einsatz unterschiedlicher Methoden ermöglichen ein besseres Verständnis des Geschehens. Neben Beobachtungsprotokollen sind auch apparative Aufnahmeverfahren hilfreich. Äußere Umstände sind ebenso zu dokumentieren, helfen sie doch bei der Interpretation des Geschehens. (Bortz & Döring, 2006, S. 340)

5. Ausstieg aus dem Feld

Nach Durchführung der Feldtätigkeit und der Dokumentation ist ein geregelter Ausstieg aus dem Feld von Nöten, der oftmals auch mit dem Auflösen von persönlichen Beziehungen zu Probandinnen und Probanden im Feld in Verbindung steht. (Bortz & Döring, 2006, S. 340)

6. Auswertung und Ergebnisbericht

Nach Verlassen des Felds beginnt die Auswertungsarbeit des umfangreichen Datenmaterials. Da die meisten Informationen in Textform vorliegen (Transkripte, Beobachtungsprotokolle, etc.) gelten die Regeln der qualitativen Auswertung. Eine der wesentlichen Entscheidungen ist, ob eine „Volltranskription“ der noch nicht in Textform vorliegenden Materialien benötigt wird, oder ob eine exemplarische Transkription relevanter Passagen ausreicht. Oftmals bieten die apparativen Aufzeichnungen die Basis für summarische Kurzbeschreibungen der Feldaktivitäten. Mit einer analytischen Darstellung wird das Datenmaterial in theoretisch sinnvolle Einheiten gegliedert, während mit einer synthetischen Darstellung Verläufe und Abläufe möglichst nachvollziehbar dargestellt werden. (Bortz & Döring, 2006, S. 340)

7.1.1 Beobachtung im Kontext der qualitativen Feldforschung

7.1.1.1 Systematische Beobachtung

Die systematische Beobachtung folgt in der Regel einem zuvor erstellten Beobachtungsplan, der die wesentlichen Fragen zur Beobachtung klären soll. Darin enthalten sind Informationen zu den Fragen: Was soll beobachtet werden? Was soll bei mehreren Beobachterinnen und Beobachtern von wem beobachtet werden? Was ist unwesentlich? Wie ist zu protokollieren? Darf interpretiert werden und wenn ja, wie? Welche Meta-Daten zur Beobachtung, wie Ort und Zeit, werden erfasst?

Eine Beobachtung wird zu einer systematischen, wenn bestimmte Ereignisse im Zentrum des Interesses liegen und Regeln festgelegt sind, die die Beobachtung nachvollziehbar machen können. (Bortz & Döring, 2006, S. 263)

7.1.1.2 Teilnehmende oder nicht-teilnehmende Beobachtung

Ist es der Forscherin oder dem Forscher möglich, selbst ein aktiver Teil des zu beobachtenden Geschehens zu sein, dann wird von einer teilnehmenden Beobachtung gesprochen. Es können somit Erkenntnisse gewonnen werden, die von außen nur schwer oder zum Teil gar nicht erkennbar sind. Es muss jedoch darauf geachtet werden, dass die Beobachterin oder der Beobachter nicht durch ihr bzw. sein aktives Verhalten das Geschehen zu sehr beeinflusst.

Der Grad der Systematisierung ist bei einer teilnehmenden Beobachtung eher gering, da es sich hierbei vorrangig um eine erkundende, explorative Tätigkeit handelt, bei der das zeitgleiche Beobachten und Dokumentieren eine Teilnahme unmöglich machen würde. Die nicht-teilnehmende Beobachtung lässt sich hingegen sehr gut systematisieren, da sich die Beobachterin oder der Beobachter vollends auf das Geschehen konzentrieren kann und nicht aktiv involviert ist. (Bortz & Döring, 2006, S. 267), (Mayring, 2002, S. 80 f.)

7.1.1.3 Offene oder verdeckte Beobachtung

Eine offene Beobachtung ermöglicht es den Feldakteurinnen und Feldakteuren, die Beobachterin und den Beobachter bewusst in ihrer bzw. seiner Rolle wahrzunehmen. Das kann dazu führen, dass die Akteurinnen und Akteure im Feld ihr Verhalten im Sinne einer „Erwünschtheit“ an die Situation anpassen. Dieses Verhalten ist jedoch nicht von allzu langer Dauer. (Bortz & Döring, 2006, S. 267 f.)

Um reaktive Ereignisse auf die Beobachtung zu vermeiden, kann eine sog. verdeckte Beobachtung durchgeführt werden. Bei dieser Art der Beobachtung wird den Akteurinnen und Akteuren im Feld nicht zur Kenntnis gebracht, dass sie beobachtet werden. Die eventuell in diesem Zusammenhang auftretenden ethischen und rechtlichen Aspekte sind entsprechend zu berücksichtigen. (Bortz & Döring, 2006, S. 268)

7.1.1.4 Apparative Beobachtung - Videographie

Unter einer apparativen Beobachtung wird der Einsatz von Ton- und Videoaufnahmegeräten zur Aufzeichnung einer Beobachtung verstanden. Diese Geräte sollen es den Beobachterinnen und Beobachtern ermöglichen, sich im Nachhinein Details der Beobachtung ansehen zu können. Gerade auch in teilnehmenden Beobachtungen erlaubt die apparative Unterstützung eine verstärkte Konzentration auf die Teilnahme, anstelle eines Hauptfokusses auf die Dokumentation der Beobachtung im Moment des Agierens im Feld. Besonders das Vorhandensein von Aufnahmegeräten im Feld kann zu reaktiven Ereignissen führen, die es entsprechend bei der Planung zu berücksichtigen gilt. (Secrist, de Koeuyer, Bell & Fogel, 2002, S. 1 ff.), (Bortz & Döring, 2006, S. 268)

Die Videographie als qualitative Forschungsmethode ähnelt sehr stark den gängigen qualitativen Vorgehensweisen. (Dinkelaker & Herrle, 2009, S. 9 & S. 17) Der große Vorteil in der apparativen Beobachtung und somit auch in der Videographie liegt in der „Haltbarkeit“ der Informationen, die einen jederzeitigen Zugriff auf das Geschehene erlaubt und somit die Beantwortung von Fragen erheblich erleichtert. (Dinkelaker & Herrle, 2009, S. 10) Die apparative Beobachtung unterstützt weiter die Objektivität der Beobachtung, da eine jederzeitige Nachverfolgbarkeit des Geschehenen gegeben ist. Die erste Interpretation bei der Protokollierung der Beobachtung durch die Beobachterinnen und Beobachter fällt weg, da die Geschehnisse mehrmals angesehen werden können. (Dinkelaker & Herrle, 2009, S. 15) Das Fehlen des unmittelbaren Erlebens der Situation im Feld für die Beobachterinnen und Beobachter des Videomaterials kann eine Limitierung der apparativen Beobachtung darstellen und kann durch eine teilnehmende Beobachtung der Beobachterinnen und Beobachter kompensiert werden. Für sie entfällt im Moment der Beobachtung die unmittelbare Dokumentationstätigkeit und sie können sich verstärkt auf das Feld und die Ereignisse im Feld konzentrieren. (Dinkelaker & Herrle, 2009, S. 16) Die Videographie eignet sich in Folge gut für eine kombinierte Erhebungsstrategie. Die erhobenen Daten können mit qualitativen Methoden ausgewertet werden, z.B. mit einer Inhaltsanalyse, einem zusammenfassenden Protokoll oder einer strukturierten Beobachtung.

7.1.2 **Generalisierbarkeit und Validität qualitativer Methoden**

In der quantitativen Forschung kann durch wahrscheinlichkeitstheoretische Methoden von Stichproben auf Populationen geschlossen werden. Hingegen können bei der qualitativen Forschung nur exemplarische Verallgemeinerungen zum Einsatz kommen, die im Idealfall „repräsentativ“ sind. Als repräsentativ gelten diese exemplarischen Verallgemeinerungen

dann, wenn sie „als typische Vertreter einer Klasse ähnlicher Fälle gelten können.“ (Bortz & Döring, 2006, S. 335) Da die qualitativen Forschungsmethoden gegenüber den quantitativen aufwändiger sind, fällt die Zahl der Probandinnen und Probanden in der qualitativen Forschung deutlich geringer aus als in der quantitativen. (Bortz & Döring, 2006, S. 335) Bortz & Döring (2006) zeigen auf, dass ein rein qualitatives Vorgehen nicht ausreichend zur Generalisierbarkeit beiträgt und somit ergänzende quantitative Verfahren eingesetzt werden sollten. (Bortz & Döring, 2006, S. 336) Die Generalisierbarkeit von Ergebnissen wird auch als externe Validität bezeichnet. Im Gegensatz dazu wird die Eindeutigkeit des Ergebnisses in Bezug auf die jeweiligen Hypothesen als interne Validität verstanden. (Bortz & Döring, 2006, S. 33)

7.2 Die qualitative Inhaltsanalyse nach Mayring

Zur Analyse und Auswertung von qualitativem Datenmaterial eignet sich unter anderem die qualitative Inhaltsanalyse nach Mayring. Diese Art der Inhaltsanalyse hat, wie auch andere qualitative Inhaltsanalysen, die Kommunikation als Untersuchungsgegenstand im Mittelpunkt, wobei die Informationen zumeist in Sprache, also Text, aber auch in Form von Ton- und Bildmaterial vorliegen. (Mayring, 2010, S. 12) Die Inhaltsanalyse kann bei einem breiten Spektrum von Materialien und Untersuchungen zum Einsatz kommen – im Gesundheitswesen, der Medienanalyse bis hin zu sozialwissenschaftlichen Fragestellungen. (Diekmann, 2009, S. 579) Die Informationen müssen protokolliert und somit fixiert sein. Die systematische Herangehensweise der Inhaltsanalyse versucht der freien Interpretation des Datenmaterials entgegenzuwirken. Man spricht hierbei auch von Regelgeleitetheit. (Mayring, 2010, S. 12)

Die Auswertung, die im Rahmen einer Inhaltsanalyse entsteht, folgt im Idealfall einer vorangehend aufgestellten Fragestellung (dieses Vorgehen ist angelehnt an eine halbstandardisierte oder standardisierte Beobachtung, siehe u.a. (Bortz & Döring, 2006, S. 270)). Sie ist daher theoriegeleitet und die Interpretation passiert vor einem theoretischen Hintergrund. Es wird versucht unter Berücksichtigung von bisherigen Erfahrungen (auch unter Bedacht des aktuellen Stands der Forschung) einen Erkenntnisfortschritt zu erreichen. (Mayring, 2010, S. 57 f.) Ein weiteres charakteristisches Merkmal der Inhaltsanalyse ist, dass es sich hierbei um eine schlussfolgernde Methode handelt, da sich die Aussagen und Auswertungen über das zu analysierende Material auf bestimmte Aspekte und Merkmale des Untersuchungsgegenstandes beziehen und somit auch Rückschlüsse zulassen. (Mayring, 2010, S. 13). Mayring (2010) schreibt, dass der Begriff der Inhaltsanalyse eher problematisch zu betrachten sei und empfiehlt die Verwendung des Begriffs „kategoriegeleitete Textanalyse“, da nicht immer nur Kommunikation und Kommunikationsverhalten Gegenstand der Analyse sind. (Mayring, 2010, S. 13)

7.2.1 Von quantitativen Auswertungsmethoden zu qualitativen

Bisher wurden in der Inhaltsanalyse vorrangig quantitative Methoden eingesetzt. (Mayring, 2010, S. 13) Eine erste, quantitative Auswertungsmethode ist die Erstellung von Häufigkeitsanalysen, im Speziellen sogenannte Frequenzanalysen. In Abhängigkeit der Fragestellung wird ein Kategoriensystem entwickelt und werden die Analyseeinheiten festgelegt. Die Kodierung resultiert in Aufzeichnungen der Häufigkeit des Auftretens von Kategorien, die anschließend berechnet, dargestellt und interpretiert werden können. (Mayring, 2010, S. 15) Der Übergang zu den qualitativen Methoden zur Analyse verlangt nach einer Berücksichtigung der Vorgehensweisen in der Forschung:

„Qualitative Wissenschaft als verstehende will also am Einmaligen, am Individuellen ansetzen, quantitative Wissenschaft als erklärende will an allgemeinen Prinzipien, an Gesetzen oder gesetzähnlichen Aussagen ansetzen. Erstere versteht sich eher als induktiv, Zweitere eher als deduktiv.“ (Mayring, 2010, S. 19)

Es kommt daher in der qualitativen Inhaltsanalyse zu einer Einzelfallorientiertheit. (Mayring, 2010, S. 20) Die qualitative Inhaltsanalyse kann, muss aber nicht, in Kombination mit einer

quantitativen Analyse auftreten. (Mayring, 2010, S. 21) Nach Mayring (2010) empfiehlt sich jedoch, die qualitative Analyse zur Fragestellung, Begriffs- und Kategorienfindung einzusetzen. Gefolgt von qualitativen oder quantitativen Analyseschritten um dann letztendlich qualitativ die Erkenntnisse auf die Fragestellung zurück zu beziehen und zu interpretieren. (Mayring, 2010, S. 21 f.)

7.2.2 Grundsätze der qualitativen Inhaltsanalyse

Die qualitative Inhaltsanalyse folgt vier Grundsätzen:

1. Systematisches Vorgehen – die qualitative Inhaltsanalyse folgt, wie auch die quantitative Inhaltsanalyse, einer Systematik.
2. Kommunikationsmodell – die qualitative Inhaltsanalyse muss das zu analysierende Material in ein Kommunikationsmodell einordnen, um den Gesamtzusammenhang nicht zu verlieren.
3. Kategorien – das Zentrum der Analyse ist, wie auch in der quantitativen Inhaltsanalyse, die Bildung und Anwendung eines Kategoriensystems.
4. Gütekriterien – die qualitative Inhaltsanalyse muss sich, wie jede andere wissenschaftliche Methode auch, an Gütekriterien messen lassen und überprüfbar sein.

(Mayring, 2010, S. 29)

Als Gütekriterien können hier die Nachvollziehbarkeit, Triangulation und Reliabilität angesetzt werden. Die Anwendung eines Ablaufmodells und das regelgeleitete Arbeiten ermöglichen die Nachvollziehbarkeit der qualitativen Inhaltsanalyse. Die Reliabilität im Sinne der „Interkoderreliabilität“ bzw. Objektivität des Kategoriensystems bezeichnet die Fähigkeit der qualitativen Inhaltsanalyse, dass unabhängige Forscherinnen und Forscher zu gleichen Ergebnissen kommen können. (Ramsenthaler, 2013, S. 25), (Diekmann, 2009, S. 593) Die Triangulation zielt darauf ab, mit unterschiedlichen Methoden und Lösungswegen zu einer Fragestellung die Stärken und Schwächen der einzelnen Methoden und Lösungswege zu kompensieren und somit ein geschlossenes Gesamtbild zu einer Fragestellung zu erstellen. (Mayring, 2002, S. 147 f.), (Diekmann, 2009, S. 543) Weitere Gütekriterien sind die Verfahrensdokumentation (Mayring, 2002, S. 144) oder auch die Nähe zum Gegenstand bzw. die Gegenstandsangemessenheit, wie nahe am Untersuchungsgegenstand geforscht wird (Feld vs. Labor). (Mayring, 2002, S. 146)

Um vor allem der Überprüfbarkeit und Nachvollziehbarkeit der Interpretation bei der qualitativen Inhaltsanalyse Rechnung zu tragen, muss das Quellmaterial und seine Entstehung hinreichend dokumentiert sein, (Mayring, 2010, S. 32) da Texte und Inhalte immer in einem Kontext interpretiert werden (Mayring, 2010, S. 48).

Ein wesentliches Merkmal der qualitativen Inhaltsanalyse ist ihre individuelle Passung, die Adäquatheit zu den konkreten Untersuchungsgegenständen. Der Gegenstand selbst steht im Vordergrund des Interesses und nicht die Technik. (Mayring, 2010, S. 50) Um die Fragestellungen der Analyse bearbeiten zu können und der Theorie zu folgen (Mayring, 2010, S. 56 ff.) ist eine spezielle Analysetechnik festzulegen und ein Ablaufmodell aufzustellen (Mayring, 2010, S. 59). Das Kategoriensystem bzw. die Kategorien zur Analyse werden in einem gegenseitigen Wechsel zwischen Theorie (Fragestellung) und Gegenstand bzw. Material entwickelt sowie durch weiterführende Konstruktions- und Zuordnungsregeln verfeinert und definiert. (Mayring, 2010, S. 59) Eine allgemeine Beschreibung eines inhaltsanalytischen Ablaufmodells findet sich bei Mayring (2010, S. 60 f.).

7.2.3 Drei Techniken zur Analyse: Zusammenfassung, Explikation und Strukturierung

Die zur Verfügung stehenden Techniken zur Analyse lassen sich zu drei wesentlichen Grundformen konsolidieren (Mayring, 2010, S. 64 f.):

- „*Zusammenfassung*: Ziel der Analyse ist es[a), Anm. d. Verf.] das Material so zu reduzieren, dass die wesentlichen Inhalte erhalten bleiben, [b), Anm. d. Verf.] durch die Abstraktion einen überschaubaren Corpus zu schaffen, der immer noch Abbild des Grundmaterials ist.
- *Explikation*: Ziel der Analyse ist es, zu einzelnen fraglichen Textteilen (Begriffen, Sätzen, ...) zusätzliches Material heranzutragen, das das Verständnis erweitert, das die Textstelle erläutert, erklärt, ausdeutet.
- *Strukturierung*: Ziel der Analyse ist es, bestimmte Aspekte aus dem Material herauszufiltern, unter vorher festgelegten Ordnungskriterien einen Querschnitt durch das Material zu legen oder das Material aufgrund bestimmter Kriterien einzuschätzen.“

(Mayring, 2010, S. 65)

7.2.3.1 Zusammenfassung

Die Grundlage für die zusammenfassende Inhaltsanalyse liefert die Psychologie der Textverarbeitung, in der die reduktiven Prozesse zur Erzeugung einer Zusammenfassung herausgearbeitet wurden. (Mayring, 2002, S. 95). Diese Prozesse werden auch Makrooperatoren genannt. (Mayring, 2010, S. 44) Die sechs Makrooperatoren lauten wie folgt:

- „*Auslassen*: Propositionen (jede bedeutungstragende Aussage, die sich aus dem Text ableiten lässt [...]), die an mehreren Stellen bedeutungsgleich auftauchen, werden weggelassen.
- *Generalisation*: Propositionen, die durch eine begrifflich übergeordnete, abstrakte Proposition impliziert werden, werden durch diese ersetzt.
- *Konstruktion*: Aus mehreren spezifischen Propositionen wird eine globale Proposition konstruiert, die den Sachverhalt als Ganzes kennzeichnet und die spezifischen Propositionen überflüssig macht.
- *Integration*: Eine Proposition, die in einer bereits durch Konstruktion gebildeten globaleren Proposition aufgeht, kann wegfallen.
- *Selektion*: Bestimmte zentrale Propositionen werden unverändert beibehalten, da sie wesentliche, bereits generelle Textbestandteile darstellen.
- *Bündelung*: Inhaltlich eng zusammenhängende, im Text aber weit verstreute Propositionen werden als Ganzes, in gebündelter Form wiedergegeben.“

(Mayring, 2002, S. 95)

Die zusammenfassende Inhaltsanalyse (siehe auch Abbildung 9) reduziert im ersten Schritt durch Paraphrasierung das Ausgangsmaterial auf eine knappe und inhaltsbeschränkte Form (Irrelevantes wird weggelassen). Dabei soll der reduzierte Inhalt auf eine einheitliche Sprachebene gebracht werden. Anschließend kommt es zur ersten Reduktion durch Selektion. Hier wird ein Abstraktionsniveau festgelegt (man spricht hier auch vom Makrooperator „Generalisation“). Alle Paraphrasen, die unter diesem Abstraktionsniveau liegen werden verallgemeinert, alle die darüberliegen werden beibehalten. Inhaltsgleiche Paraphrasen können gestrichen werden. Mit den Makrooperatoren „Auslassen“ und „Selektion“ können unwichtige und nichtssagende Paraphrasen gelöscht werden. In einem weiteren Schritt werden sich aufeinander beziehende und über das Material verteilte Paraphrasen zusammengeführt und durch neue Paraphrasen repräsentiert (Makrooperatoren „Bündelung“, „Konstruktion“ und „Integration“). Im Anschluss kann bei Bedarf eine weitere Zusammenfassung erfolgen. Bei einer großen Menge an Ausgangsmaterial können einzelne Schritte auch zusammengefasst werden. (Mayring, 2010, S. 69)

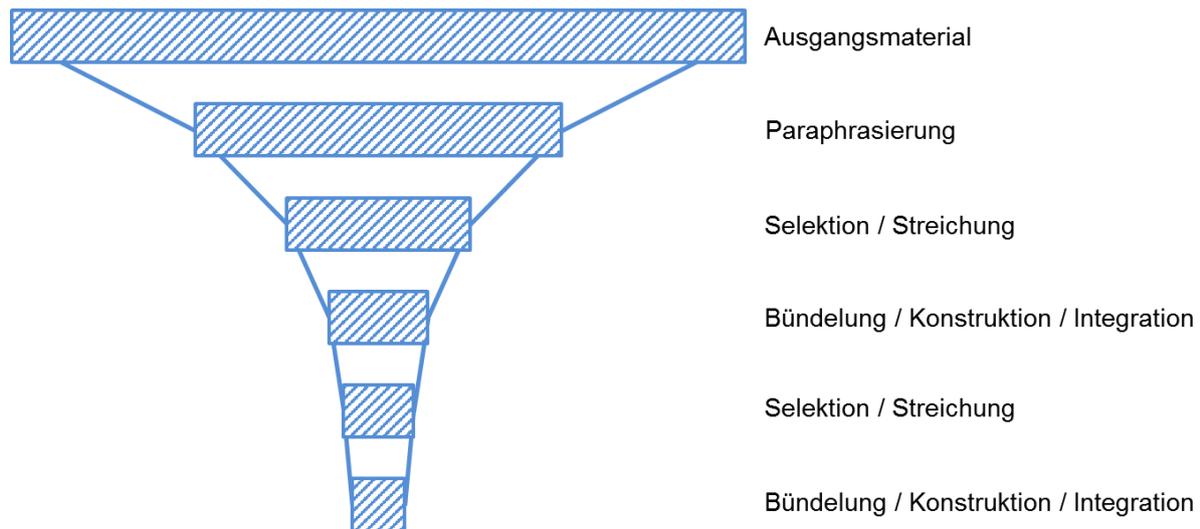


Abbildung 9: Die zusammenfassende Inhaltsanalyse

Die Definition der Kategorien kann auf Basis der zusammenfassenden Inhaltsanalyse erfolgen. Dazu kann deduktiv aber auch induktiv vorgegangen werden.

- „Eine deduktive Kategoriendefinition bestimmt das Auswertungsinstrument durch theoretische Überlegungen. Aus Voruntersuchungen, aus dem bisherigen Forschungsstand, aus neu entwickelten Theorien oder Theoriekonzepten werden die Kategorien in einem Operationalisierungsprozess auf das Material hin entwickelt. [...]
- Eine induktive Kategoriendefinition hingegen leitet die Kategorien direkt aus dem Material in einem Verallgemeinerungsprozess ab, ohne sich auf vorab formulierte Theoriekonzepte zu beziehen.“

(Mayring, 2010, S. 83)

In der Praxis werden die ersten Kategorien theoriegeleitet gebildet, im Verlauf der Inhaltsanalyse werden jedoch die Kategorien überarbeitet und nachgeschärft. Noch bevor das gesamte Material durchgearbeitet wurde, wird das adaptierte Kategoriensystem erneut auf das Material angewandt und so bestätigt oder widerlegt. Nach dem sich das Kategoriensystem gefestigt hat, wird das gesamte Material durchgearbeitet. Abschließend kann es interpretiert und analysiert werden. (Mayring, 2010, S. 84) Siehe dazu auch Huber & Gürtler (2012, S. 66 ff.).

Angelehnt an die zusammenfassende Inhaltsanalyse kann auch ein sog. zusammenfassendes Protokoll erstellt werden, welches schon bei der Erstellung die Menge an Material reduziert und konzentriert. (Mayring, 2002, S. 94 ff.)

7.2.3.2 Explikation

Da die Explikation eine untergeordnete Rolle für die gegenständliche Arbeit spielt, sei an dieser Stelle auf Mayring (2010) für eine detaillierte Darstellung der Explikation verwiesen.

7.2.3.3 Strukturierung

Die Strukturierung ist eine der zentralsten Techniken zur Inhaltsanalyse. Dabei werden systematisch die Textstellen aus dem Material herausgefiltert, die durch die jeweiligen Kategorien identifiziert und anschließend in der gewünschten Struktur abgebildet werden können. (Mayring, 2010, S. 92) Im Allgemeinen werden zu Beginn die Analyseeinheiten definiert (siehe oben), gefolgt von der theoriegeleiteten Festlegung der Strukturierungsdimensionen. (Mayring, 2010, S. 93) Die Strukturierungsdimensionen werden meist noch detaillierter betrachtet und in zusätzliche Ausprägungen zerlegt. So entsteht ein theoriegeleitetes Kategoriensystem. Bei der Kategorienfindung werden Textstellen definiert,

die in die jeweilige Kategorie zuzuordnen sind. Um die Kodierung zu erleichtern, werden sog. Ankerbeispiele entwickelt – das sind konkrete Textstellen, die beispielhaft genannt werden. Zusätzliche Kodierregeln sollen die Abgrenzung bei der Zuordnung zu den Kategorien unterstützen. (Mayring, 2010, S. 92) Im nächsten Schritt werden die Fundstellen im Material, akkordierend zum Kategoriensystem, gekennzeichnet und in weiterer Folge extrahiert und bearbeitet. Allenfalls kommt es zu einer Adaption des Kategoriensystems und seiner Regeln. Zum Abschluss werden die Ergebnisse aufbereitet, interpretiert und ausgewertet. (Mayring, 2010, S. 93)

Die Strukturierung kann nach vier Gesichtspunkten erfolgen: nach formalen, nach inhaltlichen, nach typisierenden und nach skalierenden. Bei der formalen Strukturierung wird versucht eine Struktur im Material herauszuarbeiten, die das Material untergliedert oder schematisiert. Es können somit thematische Einheiten, Argumentationsstrukturen oder Gesprächsstrukturen ermittelt werden. (Mayring, 2010, S. 94) Bei der inhaltlichen Strukturierung werden bestimmte Themen, Inhalte oder Aspekte herausgefiltert und zusammengefasst. Es gelten hierbei die Regeln der Zusammenfassung (siehe oben). Die typisierende Strukturierung konzentriert sich auf besonders bedeutungsvolle und markante Gegenstände. Dazu können Typen unterschiedlichster Dimensionen verwendet werden, nicht nur Menschen. Um die Typisierung durchführen zu können, müssen die Extreme, „die Ausprägungen von besonderem theoretischem Interesse“ und besonders häufig vorkommende Ausprägungen der Typen beschrieben werden. (Mayring, 2010, S. 98)

Die skalierende Strukturierung hat zum Ziel, Merkmale und Ausprägungen von Material auf einer Skala zu positionieren. Hierzu können die Ansätze der quantitativen Inhaltsanalyse gut verwendet werden, jedoch muss das Zentrum der Analyse die Kategorienbildung und Zuordnung sein. Als Strukturierungsdimensionen kommen nun Einschätzungsdimensionen – Variablen mit Ausprägungen entlang einer Skala, zumindest einer Ordinalskala – zum Einsatz. (Mayring, 2010, S. 101) Die Einschätzungsdimensionen werden theoriegeleitet aus der Fragestellung abgeleitet. (Mayring, 2010, S. 103) Bei der Bearbeitung des Materials wird selbiges diesen Skalen zugeordnet. Zur Ergebnisfindung werden diese Zuordnungen wiederum zusammengefasst und meistens quantitativ analysiert. (Mayring, 2010, S. 101)

In Anlehnung an die strukturierte Inhaltsanalyse kann ebenso eine strukturierte Beobachtung durchgeführt werden. Auch hier wird ein Plan, ein sog. Beobachtungsplan, entwickelt und mit einem Beobachtungsschema der Beobachtungsgegenstand strukturiert erfasst und dokumentiert. (Diekmann, 2009, S. 569 ff.)

8 Empirische Entwicklung eines Frameworks

Zur Überprüfung der Hypothese, ob 3- bis 6-jährige Kinder, nachdem sie altersgerecht in der Programmierung einfacher, grundlegender Befehlsabfolgen geschult wurden, Problemstellungen desselben Schwierigkeitsgrades nach Teilnahme an Aktivitäten des Programms selbständig lösen können, führte die Autorin entsprechende Lehr-Lern-Einheiten zur frühen Technikbildung in Informatik durch und leitete daraus ein wissenschaftlich und theoretisch fundiertes, empirisch belegtes sowie kompetenzbasiertes Framework ab.

Für die Entwicklung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ holte sich die Verfasserin der gegenständlichen Arbeit Anregungen und Inspirationen u.a. bei dem CSTA-Framework Level 1 (siehe Kapitel 3.1.2), den „four P's of Creative Learning“ von Resnick (2014) (siehe Kapitel 3.1.2), dem sechs C-Modell (siehe Kapitel 3.1.2), den Grundlagen informatischer Bildung nach Hubwieser (2007) (siehe Kapitel 3.1), den Bildungswerten der Informatik nach Futschek (2003) (siehe Kapitel 3.1.2) und dem PLU-Modell von Markopoulos et al. (2008) (siehe Kapitel 5.1.1.1). Inquiry Based Learning kommt neben Discovery Learning (beide siehe Kapitel 2.3.1) im Framework als Integrationsmethode von IKT in Lehr-Lern-Szenarien zum Einsatz. Details hierzu sind Kapitel 8.2 zu entnehmen.

Ziel des Frameworks ist es eine kompetenzbasierte Möglichkeit zur frühen Technikbildung in Informatik für 3- bis 6-jährige Kinder zur Verfügung zu stellen. Die Ableitung von Aufgaben aus den im Framework enthaltenen Aktivitätsschablonen hilft lernergebnisorientiert Kindern informatische Themen zu vermitteln. Darüber hinaus unterstützt das Framework bei der Ausgestaltung der abgeleiteten Aufgaben, da bei jeder Aktivitätsschablone zusätzlich zur Aufgabenstellung weitere Informationen wie Sozialform, einzusetzende Materialien oder Hinweise zur Durchführung enthalten sind.

Der Nutzen des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ liegt in seinem Einsatz im Kindergarten zur frühen Technikbildung in Informatik begründet. Personen, die frühe Technikbildung in Informatik im Kindergarten betreiben möchten, wird hiermit ein Hilfsmittel zur Verfügung gestellt, mit dem individuelle Aufgaben zur fortwährenden aktiven Auseinandersetzung der Kinder mit Informatik abgeleitet werden können. Die entwickelten Aktivitätsschablonen bieten eine breite Palette an Aktivitäten rund um Informatik an, die nicht nur das Thema Programmierung abdecken, sondern in drei Leitthemen („Early Childhood Technology“, „Computational Thinking“, „Child-Centered-Design“) die Vielfältigkeit der Informatik behandeln.

Die Innovation des Frameworks ist, dass vier Perspektiven (fokussiert auf die Leitthemen, die Lernergebnisse, die Tätigkeiten oder die Inhaltsdimensionen) helfen beliebig viele, auch zeitlich kürzere aber immer lernergebnisorientierte, Aufgaben zu erstellen. Diese können in Lehr-Lern-Einheiten zusammengefasst werden und fortwährend, über einen längeren Zeitraum in die Arbeit mit Kindergartenkindern einfließen. Weiter können mit den vier Perspektiven unterschiedliche Bedürfnisse und Herangehensweisen in der Wissensvermittlung, unabhängig von der eingesetzten Hard- und Software, optimal unterstützt werden. Alle vier Perspektiven dienen als Ausgangspunkte zur Ableitung von Aufgaben aus den Aktivitätsschablonen.

In der ersten Perspektive können lernergebnisorientierte Aufgaben aus den Aktivitätsschablonen entlang der Inhaltsdimensionen abgeleitet werden. In der zweiten Perspektive sind die Aktivitätsschablonen nach Leitthemen organisiert. Die Lernergebnisse selbst stehen in der dritten Perspektive im Fokus des Interesses – aus welchen Aktivitätsschablonen empfiehlt es sich Aufgaben abzuleiten um ein bestimmtes Lernergebnis zu unterstützen? Die vierte Perspektive zeigt die Aktivitätsschablonen den eingesetzten Tätigkeiten und den damit verbundenen Lernergebnissen zugeordnet.

Die Kapitel 8.1 bis 8.6 beschreiben die Entwicklung des Frameworks. Kapitel 8.6.2 präsentiert empirische Erkenntnisse über die erzielten Lernergebnisse der Kinder, die dem Framework zugrundeliegende konkrete Aufgaben bearbeitet haben. In Kapitel 8.7 stellt die Autorin das finale Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ vor und zeigt

anhand konkreter Beispiele den Einsatz des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ in Kapitel 8.8.

8.1 Prozess zur Entwicklung des Frameworks

Die Verfasserin der gegenständlichen Arbeit entwickelte induktiv aus zwei konkreten und empirisch befundeten Durchführungen von Lehr-Lern-Einheiten in zwei Kindergärten das verallgemeinerte Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ in 12 Prozessschritten, die in sechs Phasen unterteilt sind.

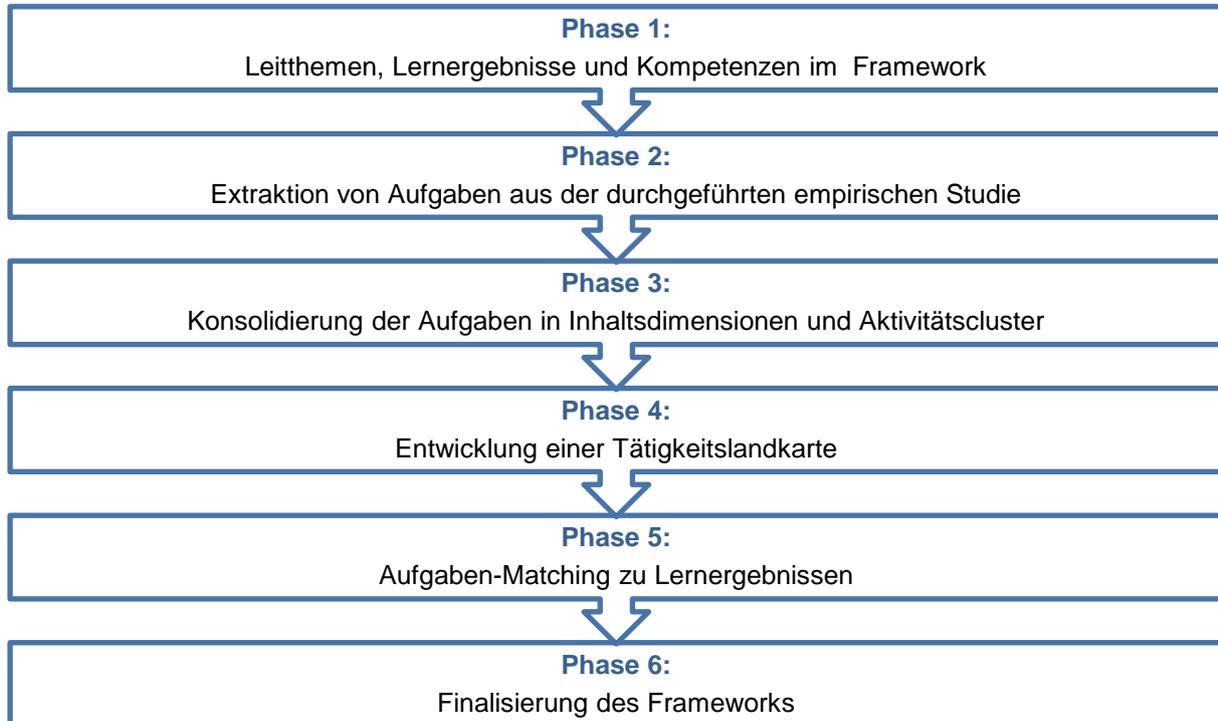


Abbildung 10: Phasen des Prozesses zur Entwicklung des Frameworks

Phase 1: Leitthemen, Lernergebnisse und Kompetenzen im Framework (siehe Kapitel 8.2)

- Schritt:**
Zu Beginn identifizierte die Autorin aus der Literatur (siehe Kapitel 2 bis Kapitel 5) drei Leitthemen für das Framework (siehe Kapitel 8.2.1 bis Kapitel 8.2.3).
- Schritt:**
Basierend auf der facheinschlägigen Literatur (siehe Kapitel 3.2) erfolgte die Festlegung der durch die Leitthemen zu erzielenden Kompetenzen. (siehe Kapitel 8.2.4)
- Schritt:**
Aus der eigenen Erfahrung aus Vorarbeiten zur gegenständlichen Arbeit und aus der Literatur leitete die Autorin die Lernziele zu den Leitthemen und in Korrelation dazu die Lernergebnisse entlang der zu erzielenden Kompetenzen ab. (siehe Kapitel 8.2.1, 8.2.2, 8.2.3 sowie 8.2.4)
- Schritt:**
Zum Abschluss der Phase 1 folgte die Entwicklung von Aufgabenentwürfen entlang der Leitthemen und der Lernziele sowie der Lernergebnisse. (siehe Kapitel 8.2.5)

Phase 2: Extraktion von Aufgaben aus der durchgeführten empirischen Studie (siehe Kapitel 8.3)

- Schritt:**
Die Durchführung der Aufgabenentwürfe in zwei Kindergärten im Rahmen eines Programms zur frühen Technikbildung in Informatik dokumentierte die Verfasserin der

gegenständlichen Arbeit mit apparativer Beobachtung, um diese anschließend unter Einsatz von Methoden der Inhaltsanalyse auswerten zu können. (siehe Kapitel 8.3.1 und 8.3.2)

6. Schritt:

Im nächsten Schritt transkribierte und kodierte die Autorin das Beobachtungsmaterial (siehe Kapitel 8.3.3) und führte eine Inhaltszusammenfassung durch. (siehe Kapitel 8.3.4)

7. Schritt:

Aus der Inhaltszusammenfassung folgte eine Extraktion von Meta-Daten zu den tatsächlich durchgeführten Aufgaben, wie eingesetzte Sozialform, Schwierigkeitsgrad und Art der Aufgabe. (siehe Kapitel 8.3.5)

Phase 3: Konsolidierung der Aufgaben in Inhaltsdimensionen und Aktivitätscluster (siehe Kapitel 8.4)

8. Schritt:

Im Folgenden konsolidierte die Autorin die Aufgaben in Inhaltsdimensionen (siehe Kapitel 8.4.1) und innerhalb der Inhaltsdimensionen wiederum in Aktivitätscluster (siehe Kapitel 8.4.2).

Phase 4: Entwicklung einer Tätigkeitslandkarte (siehe Kapitel 8.5)

9. Schritt:

Basierend auf Schritt 7 erfolgte eine Extraktion der eingesetzten Tätigkeiten inkl. deren Unterteilung in Lehr-, Haupt- und Unterstützungsaktivitäten. Anschließend entwickelte die Verfasserin der gegenständlichen Arbeit eine Tätigkeitslandkarte. (siehe Kapitel 8.5)

Phase 5: Aufgaben-Matching zu Lernergebnissen (siehe Kapitel 8.6)

10. Schritt:

Matching der bisherigen Aufgaben hinsichtlich der zu erwartenden Lernergebnisse. (siehe Kapitel 8.6.1)

11. Schritt:

Exemplarische Überprüfung der Aufgaben in Bezug auf die Erreichung der jeweiligen Lernergebnisse unter Einsatz eines Abschluss-Assessments (siehe Kapitel 8.6.2)

Phase 6: Finalisierung des Frameworks (siehe Kapitel 8.7)

12. Schritt:

Im letzten Schritt leitete die Autorin aus den bestehenden Aufgaben verallgemeinerte Aktivitätsschablonen ab und entwickelte somit ein generell einsetzbares Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zur frühen Technikbildung in Informatik. (siehe Kapitel 8.7)

8.2 Phase 1: Leitthemen, Lernergebnisse und Kompetenzen im Framework

Das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ besteht aus drei Leitthemen: „Early Childhood Technology“ (siehe Kapitel 8.2.1), „Computational Thinking“ (siehe Kapitel 8.2.2), und „Child-Centered-Design“ (siehe Kapitel 8.2.3). Die Grundlagen zur Ausprägung der Leitthemen entstammen zum einen der Literaturrecherche, dargelegt in den Kapiteln 2 bis 5, und zum anderen folgenden konkreten Anregungen:

Das Level 1 des CSTA-Frameworks (siehe Kapitel 3.1.2) liefert mit dem Bereich „Computational Thinking“ nicht nur einen erheblichen Input für das Leitthema „Computational Thinking“, sondern unterstützt den Inhalt des gegenständlich zu entwickelnden Frameworks unter anderem auch dahingehend, dass Technologie genutzt werden kann, um altersgerechte Probleme zu lösen oder aber auch dass der Einsatz von Digitalkameras zu Dokumentationszwecken in der Forschung erfolgen kann. Die Vermittlung der Tatsache, dass Software entwickelt wird, um Computer oder Roboter zu steuern, dass dafür sogar ein

eigenes Befehlset zur Lösung einfacher Aufgaben genutzt werden kann, und dass dies kooperativ und kollaborativ mit Lehrpersonen und Peers erfolgt, sind weitere Grundideen, auf denen aufgebaut wurde.

Die vier Ps (Projekte, Peers, Passion, Play) von Resnick (2014) (siehe Kapitel 3.1.2) inspirierten zur Gestaltung von Aufgaben und insbesondere zur Wahl von Sozialformen, Tätigkeiten und Materialien.

Zur Ausgestaltung von Ideen für Aufgaben und Leitthemen wurde das sechs C-Modell (siehe Kapitel 3.1.2) herangezogen, wobei hier vorrangig die weitere Verfeinerung, das „C's by six C's model“ einen Beitrag leistet: Kollaborationen, Kompetenzen und Kreativität spielen eine wesentliche Rolle für die Gestaltung des gegenständlich zu entwickelnden Frameworks.

Den Grundlagen informatischer Bildung nach Hubwieser (2007) (siehe Kapitel 3.1) folgend, werden die eingesetzten technischen Materialien, der Bodenroboter und die Simulation des Bodenroboters, zur Unterstützung von Lernvorgängen, bei denen die Beherrschung grundlegender informatischer Konzepte vermittelt werden, betrachtet. Bedienerfertigkeiten werden anhand von problembasierten Aufgabenstellungen trainiert.

Aus den Bildungswerten der Informatik nach Futschek (2003) (siehe Kapitel 3.1.2) abgeleitet, werden im gegenständlichen Framework im Speziellen Aufgaben zum algorithmischen Denken entwickelt und das kooperative, interdisziplinäre Arbeiten gefördert.

Die Autorin achtete bei der Gestaltung des gegenständlich zu entwickelnden Frameworks gemäß dem PLU-Modell von Markopoulos et al. (2008) (siehe Kapitel 5.1.1.1) darauf, dass Kinder in mehreren Rollen – als Spielende, Lernende und als Benutzerinnen und Benutzer von Produkten – aktiv an technologiebezogenen Lehr-Lern-Aktivitäten partizipieren.

Bei der Entwicklung des gegenständlichen Frameworks berücksichtigt die Autorin die Methodik des Inquiry Based Learning (siehe Kapitel 2.3.1). Lernende können hierbei Fragen und Probleme aufwerfen, die kollaborativ im Team bearbeitet werden. Ergänzt wird dies durch das Discovery Learning (siehe Kapitel 2.3.1), bei dem die Lernenden durch Interaktionen mit der Umwelt und durch Manipulation von Objekten sowie durch Experimente lernen.

Zu den entwickelten Leitthemen leitete die Verfasserin der gegenständlichen Arbeit jeweils geeignete Lernziele und Lernergebnisse ab. Letztere ordnete die Autorin im Anschluss den zuvor festgelegten Kompetenzen zu (siehe Kapitel 8.2.4).

8.2.1 Leitthema: Early Childhood Technology

8.2.1.1 Lernziele

- Z1.1. Die Kinder sollen einen ersten Einblick in die Welt der Roboter bekommen. Die Neugierde für Robotik soll bei den Kindern durch lockere Gespräche geweckt werden.
- Z1.2. Schon vorhandenes Wissen zur Robotik soll spielerisch aktiviert werden (Roboter im Fernsehen, Sinnesorgane für die Interaktion mit der Welt, Strom als „Essen“, etc.).
- Z1.3. Die Kinder sollen einen ersten Eindruck davon erhalten, welche Alltagsaufgaben in ihrem unmittelbaren Umfeld bereits von Robotern übernommen werden können. Dazu wird z.B. ein Staubsauger-Roboter demonstriert, der wie ein flacher Zylinder aussieht, aus Kunststoff besteht und keine Ähnlichkeit mit einer „Blechbüchse“ oder einem humanoiden Roboter aufweist.
- Z1.4. Die Kinder sollen Roboter kennenlernen, die nicht in ihrem unmittelbaren Umfeld zum Einsatz kommen, wie z.B. Fußball- oder Industrie-Roboter.

- Z1.5. Die Kinder sollen eine einfache Art zur Dokumentation von Ereignissen kennenlernen, um sich länger an diese erinnern zu können.
- Z1.6. Die Kinder sollen erfahren, wie Foto-Kameras funktionieren, wie man sie bedient und wie Fotos „entwickelt“ werden. Dazu werden Kinderkameras und portable Fotodrucker eingesetzt, die an die Kameras direkt angeschlossen werden können.
- Z1.7. Die Kinder sollen den Einsatz von Fotokameras als Dokumentationsmethode erleben, vertiefen und nutzen.
- Z1.8. Die Kinder sollen ihr Forschungsheft als chronologische Dokumentation des Programms verstehen und den Datumsbegriff kennenlernen oder auffrischen.
- Z1.9. Die Kinder sollen üben Inhalte zusammenzufassen, indem sie die Programminhalte rückblickend wiedergeben.

8.2.1.2 Lernergebnisse

Nach Besuch der Lehr-Lern-Einheiten sind die Kinder in der Lage, ...

- LE1.1. technische Begriffe in eigenen Worten zu erklären.
- LE1.2. ein Praxisbeispiel zum Einsatz von Robotern im Alltag zu nennen und die Funktionsweise desselben in eigenen Worten zu erläutern.
- LE1.3. ein sich bewegendes Objekt, z.B. einen Staubsauger-Roboter, zu fotografieren.
- LE1.4. Roboter, die nicht in ihrem alltäglichen Umfeld vorkommen, zu benennen und zu dokumentieren.
- LE1.5. mit Hilfe ihres Forschungshefts in eigenen Worten zu erklären, was unter dokumentieren zu verstehen ist.
- LE1.6. über einen längeren Zeitraum Erlebtes und Erlerntes zusammenfassend wieder zu geben.
- LE1.7. (früher) erworbenes themenrelevantes Wissen abzurufen und zur Lösung neuer Aufgabenstellungen anzuwenden.

8.2.2 Leitthema: Computational Thinking

8.2.2.1 Lernziele

- Z2.1. Kinder lernen Bodenroboter, die von ihnen selbst programmiert werden können, zum Beispiel den Bee-Bot, kennen.
- Z2.2. Die Kinder sollen eigenständig mit dem Bodenroboter ihnen gestellte Programmieraufgaben verschiedenen Schwierigkeitsgrades lösen.
- Z2.3. Kinder lernen die Simulation eines Bodenroboters, die von ihnen selbst programmiert werden kann, zum Beispiel des Bee-Bots, kennen.
- Z2.4. Die Kinder sollen eigenständig mit der Simulation eines Bodenroboters üben und ihnen gestellte Programmieraufgaben lösen.
- Z2.5. Die Kinder sollen in einem Wettbewerb unterschiedliche Lösungsstrategien für Programmieraufgaben kennenlernen.
- Z2.6. Die Kinder sollen anderen Kindern Aufgaben stellen, die mit dem Bodenroboter gelöst werden können.
- Z2.7. Die Kinder sollen sich gegenseitig beim Lösen von Aufgaben unterstützen.
- Z2.8. Die Kinder sollen spielerisch in einem Wettrennen unter Zeitdruck ihre bis dahin erworbenen Programmierfähigkeiten einsetzen.
- Z2.9. Das Ausprobieren und das Lernen durch Versuch und Irrtum sollen den Entdeckergeist der Kinder anregen, analytisches Schließen fördern und das Interesse für technische Fragestellungen wecken bzw. fördern.

8.2.2.2 Lernergebnisse

Nach Besuch der Lehr-Lern-Einheiten sind die Kinder in der Lage, ...

- LE2.1. einfache und komplexe Programmier-Probleme zu lösen.
- LE2.2. informatische Begriffe in eigenen Worten zu erklären.
- LE2.3. einen komplexen Weg in Einzelschritte zu zerlegen.
- LE2.4. Ursache-Wirkungs-Prinzipien zu erkennen.
- LE2.5. die grundlegende Funktionalität eines Bodenroboters, der über eine Mini-Language verfügt, zum Beispiel anhand des Bee-Bots, zu erklären.
- LE2.6. einen Bodenroboter zur Lösung von Aufgabenstellungen sinngemäß zu bedienen.
- LE2.7. anderen Kindern einfache Aufgabenstellungen mit dem Bodenroboter zu stellen.
- LE2.8. Aufgabenstellungen anderer Kinder selbständig (ohne Hilfe von erwachsenen Betreuungspersonen) mit dem Bodenroboter zu lösen.
- LE2.9. die grundlegende Funktionalität der Simulation eines Bodenroboters in dessen Mikrowelt, zum Beispiel korrespondierend zum Bee-Bot, zu erklären.
- LE2.10. die Simulation eines Bodenroboters zur Lösung von Aufgabenstellungen in dessen Mikrowelt sinngemäß zu bedienen.
- LE2.11. sich im Raum zu orientieren, die Richtungen zu benennen und anzuwenden.
- LE2.12. unterschiedliche Perspektiven einzunehmen.
- LE2.13. gemeinsam an der Lösung von Aufgaben zu arbeiten.
- LE2.14. geeignete Anweisungen oder Hilfestellungen zu geben.

8.2.3 Leitthema: Child-Centered-Design

8.2.3.1 Lernziele

- Z3.1. Das Zeichnen eines eigenen Roboters und die Erklärungen sollen den Kindern helfen eigene Vorstellungen auszudrücken sowie Gedanken in Bilder und Worte zu fassen.
- Z3.2. Die Kinder sollen im Team eigene Materialien zur späteren Nutzung mit dem Bodenroboter zeichnerisch gestalten.
- Z3.3. Die Kinder sollen Prototypen von Bodenrobotern gestalten.
- Z3.4. Die Kinder sollen eine Benutzerschnittstelle einer Mikrowelt evaluieren.
- Z3.5. Die Kinder sollen einen Bodenroboter bezüglich seiner Usability evaluieren.

8.2.3.2 Lernergebnisse

Nach Besuch der Lehr-Lern-Einheiten sind die Kinder in der Lage, ...

- LE3.1. Prototypen eines Bodenroboters zu gestalten.
- LE3.2. selbsterstellte Artefakte zu präsentieren
- LE3.3. Bodenroboter-Matten zeichnerisch im Team zu gestalten.
- LE3.4. eine offene Fragestellung zu beantworten bzw. einer offenen Aufgabenstellung zu folgen.
- LE3.5. das eigene Handeln zu kommentieren (Talk Aloud, siehe Kapitel 5.3.1).

8.2.4 Zuordnung Lernergebnisse zu Kompetenzen

Basierend auf der Darlegung möglicher zu erwerbender Kompetenzen durch den Einsatz von IT und Robotern zur frühen Technikbildung, siehe Kapitel 3.2, hat die Verfasserin der gegenständlichen Arbeit die für die drei Leitthemen (siehe Kapitel 8.2.1, 8.2.2, 8.2.3) aus

ihrer Sicht relevanten Kompetenzen herausgefiltert und bestimmt. Der Clusterung von Eguchi (2007) folgend, teilte die Autorin die Kompetenzen zum einen in Fachkompetenzen (FK) und zum anderen in akademische Kompetenzen (AK) ein. Eine Zuordnung der Lernergebnisse der drei Leitthemen zu den jeweiligen Kompetenzen ist der folgenden Tabelle zu entnehmen.

	Informatik, FK	Technik allgemein, FK	Mathematik, FK	Gestalten, AK	Dokumentieren, AK	Reflektieren, AK	Kooperativ Arbeiten, AK	Kreativ Denken, AK	Entscheidungen treffen, AK	Probleme lösen, AK	Kommunizieren, AK	Präsentieren, AK	Räumlich Denken, AK
LE1.1		x									x		
LE1.2		x									x		
LE1.3		x			x								
LE1.4		x			x						x		
LE1.5					x						x		
LE1.6						x					x	x	
LE1.7						x		x	x	x			
LE2.1	x		x					x	x	x			
LE2.2	x										x	x	
LE2.3			x						x	x			x
LE2.4	x	x				x		x					
LE2.5	x	x									x	x	
LE2.6	x		x					x	x	x			x
LE2.7	x							x			x		
LE2.8	x	x	x					x	x	x			x
LE2.9	x	x									x	x	
LE2.10	x	x	x					x		x			x
LE2.11									x				x
LE2.12								x	x	x			x
LE2.13					x	x	x	x	x	x	x		
LE2.14						x	x			x	x		
LE3.1	x	x		x				x		x			x
LE3.2						x						x	
LE3.3				x			x						
LE3.4								x	x	x			
LE3.5						x					x		

Tabelle 2: Zuordnung Lernergebnisse zu Kompetenzen

8.2.5 Entwicklung von Aufgabenentwürfen

Die Auswahl der Aufgabenentwürfe orientierte sich zum einen an den Darlegungen zur Kompetenzentwicklung durch Einsatz von IT und Robotern (siehe Kapitel 3.2), zum anderen an der kognitiven Entwicklung nach Piaget (siehe Kapitel 2.1.3) sowie an weiteren Erkenntnissen zur kognitiven und motorischen Entwicklung von Kindern der Zielgruppe (siehe Kapitel 5.1.1). Besonderes Augenmerk schenkte die Autorin dem algorithmischen

Denken (siehe Kapitel 3.1.4), dem Higher-Order-Thinking und dem Computational Thinking (siehe Kapitel 3.3), zu denen sich im Bereich des Leitthemas „Computational Thinking“ Aktivitäten finden.

In weiterer Folge leitete die Verfasserin der gegenständlichen Arbeit entlang der zuvor definierten Leitthemen, der Lernziele und der Lernergebnisse Aufgabenentwürfe zu folgenden inhaltlichen Schwerpunkten ab:

- Einstieg ins Thema Roboter, Vorstellung der handelnden Personen, Einführung Forschungsheft
- Kinder zeichnen einen Roboter und präsentieren diesen anschließend
- Usability-Test Bodenroboter, jedes Kind einzeln
- Erklärung des Bodenroboters
- Programmieren von Bodenrobotern zum Fahren (auch auf vorgefertigten Matten)
- Lernen von Links- und Rechts-Drehungen mit dem Bodenroboter
- Vorstellung der digitalen Fotoapparate für Kinder und eines Staubsaugerroboters
- Erstellen von Fotos mit den Fotoapparaten für Kinder
- Gestalten von eigenen Materialien für die Verwendung mit den Bodenrobotern
- Einkleben von Roboterbildern und selbstgemachten Fotos vom Staubsaugerroboter ins Forschungsheft
- Einen Roboter-Prototyp zeichnen / entwerfen und anschließend basteln
- Vorstellung der von den Kindern gebastelten Roboter-Prototypen vor der Gruppe
- Usability-Test Softwaresimulation des eingesetzten Bodenroboters, jedes Kind einzeln
- Programmieren der Softwaresimulation des eingesetzten Bodenroboters
- Roboter Theater spielen: Roboter spielen, Befehle geben, Befehle ausführen
- Abschluss-Assesement, jedes Kind einzeln

Ein Aufgabenentwurf ist hierbei eine abstrakte, stichwortartige, informelle Zusammenfassung über den im Lehr-Lern-Setting umzusetzenden Lehr-Lern-Inhalt.

8.2.6 Exemplarisch eingesetzte Hard- und Software

In der gegenständlichen Forschungsarbeit leitete sich zum einen aus der Theorie (siehe Kapitel 4.4) und zum anderen aus den Vorarbeiten und -erfahrungen der Autorin (siehe Kapitel 1.4 und 1.6) die Auswahl des Bee-Bots, eines einfachen Bodenroboters der Firma TTS Group Ltd, zur exemplarischen Nutzung ab. Als Softwaresimulation des Bodenroboters kamen die korrespondierenden Produkte der Firma, Focus on Bee-Bot 1 und Focus on Bee-Bot 2, (siehe Kapitel 4.3.1.3.3) zum Einsatz.

Der Bee-Bot verfügt über ein tangible User Interface (siehe Kapitel 4.3.2 und 4.4.2) und nutzt die gleiche Mini-Language (siehe Kapitel 4.3.1.2), die in der Software (Mikrowelt, siehe Kapitel 4.3.1.1) Focus on Bee-Bot 1 und 2 genutzt wird. Beide (Hard- und Software) agieren unabhängig voneinander, jedoch erfolgt bei den Kindern ein Wissenstransfer (siehe Kapitel 9) zwischen den beiden Varianten zum Programmieren lernen, wenn beide zum Einsatz kommen.

Die eingesetzte Hard- und Software wurde im Rahmen der Studie auf ihre jeweilige Usability und Eignung zur frühen Technikbildung in Informatik für 3- bis 6-jährige Kinder evaluiert (siehe Kapitel 9).

Der ausgewählte Bodenroboter und die ausgewählte Software sind als Beispiele zur Nutzung mit dem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zu verstehen. Das

finale Framework ist herstellerunabhängig und kann mit beinahe beliebiger, zielgruppengeeigneter Hard- und Software genutzt werden.

8.3 Phase 2: Extraktion von Aufgaben aus der durchgeführten empirischen Studie

Zur empirischen Evaluierung und Überprüfung der entwickelten Aufgabenentwürfe hielt die Autorin zwei exemplarische Programme, in zwei unterschiedlichen Kindergärten in Steyr, Oberösterreich (fortan Kindergarten D und E genannt) ab. Die zwei Programme hatten je einen Umfang von fünf Lehr-Lern-Einheiten zu einer Dauer à drei Stunden. Es nahmen pro Kindergarten / pro Durchführung des Programms je zehn Kinder (fünf Mädchen, fünf Jungen) im Alter von 3 bis 6 Jahren teil. Im Fokus der Durchführung der Programme standen das Erfassen und Dokumentieren der Reaktionen der Kinder auf die tatsächlich gestellten Aufgaben (welche auf den zuvor erstellten Aufgabenentwürfen basierten) um evaluieren zu können, welche Aufgaben tatsächlich mit Kindern im Alter von 3 bis 6 Jahren im Kontext einer frühen Technikbildung in Informatik durchgeführt werden können und welche nicht. In weiterer Folge soll ein abstrahiertes Framework mit einem Set von verallgemeinerten Aktivitätsschablonen entstehen, welche zur frühen Technikbildung in „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zum Einsatz kommen können.

8.3.1 Dokumentation

Für diese Forschungsarbeit war die Nachvollziehbarkeit der durchgeführten Aufgaben ein Anliegen und somit zentrales Thema der Studie. Dazu zeichnete das Unterstützungspersonal der Autorin alle Lehr-Lern-Einheiten der Programme, das entspricht der gesamten Anwesenheit der Autorin und ihrer Unterstützer in den Kindergärten, apparativ auf. Nur so konnte sichergestellt werden, dass die gewonnenen Erkenntnisse korrekt wiedergegeben und valide Ergebnisse abgeleitet werden können. Im Vorfeld klärte die Verfasserin der gegenständlichen Arbeit die teilnehmenden Kinder, deren Eltern sowie das Betreuungspersonal im Kindergarten darüber auf, dass die apparative Aufzeichnung ausschließlich zu wissenschaftlichen Zwecken erfolgte. Zur rechtlichen Absicherung unterschrieben alle Erziehungsberechtigten eine Freigabe des Rechts am Bild ihrer Kinder im Rahmen der gegenständlichen Forschungsarbeit. Die Autorin sicherte zu, dass die Daten nicht öffentlich zugänglich gemacht werden. Die Anonymisierung der Namen der Kinder und die Unkenntlichmachung von Gesichtern in Publikationen zu dieser Forschungsarbeit, einschließlich der gegenständlichen Arbeit, vereinbarten alle Beteiligten als obligatorisch.

Da die gesamte inhaltsbezogene Aufenthaltsdauer der Verfasserin der gegenständlichen Arbeit in den jeweiligen Kindergärten vollinhaltlich videotechnisch erfasst wurde, mussten sich die Autorin und das Unterstützungspersonal nicht um die Dokumentation der Geschehnisse annehmen, sondern konnten sich voll und ganz dem Inhalt widmen. Die Video-Aufzeichnung erfolgte mit zwei Kameras aus unterschiedlichen Blickwinkeln. Die Kameras waren in den Raumecken etwas abseits des Geschehens positioniert, so dass weder die Kinder noch die Autorin bei der Durchführung der Lehr-Lern-Einheiten durch die Geräte gestört wurden. Zusätzlich zur Aufzeichnung mit Video dokumentierte das Unterstützungspersonal das Geschehen auch mit einem Audio-Aufnahmegerät. Einerseits als Backup-Lösung, sollten die Videokameras ausfallen und andererseits als Unterstützung für die Videokameras, um den Ton im Raum besser erfassen zu können (siehe dazu Abbildung 11).

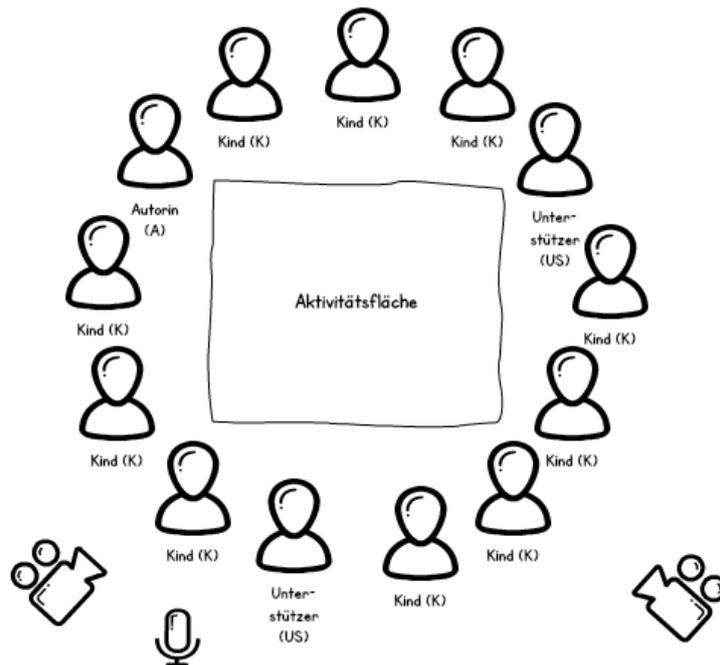


Abbildung 11: Schematische Darstellung Dokumentation des Programms

An den beiden Programmen nahmen jeweils bis zu zehn Kinder teil – eine Schwankung der Teilnehmerzahl ergab sich durch unregelmäßige Anwesenheit der Kinder, obwohl die Lehr-Lern-Einheiten immer an fünf aufeinanderfolgenden Tagen stattfanden.

Die Autorin leitete die Lehr-Lern-Einheiten und vermittelte den Kindern maßgeblich die entsprechenden Inhalte. Das Unterstützungspersonal, hier zwei Personen, war anwesend, um die Dokumentation technisch einwandfrei abzuwickeln und bei der Betreuung der Kinder, vor allem in den jeweiligen Einzelsettings, mitzuwirken.

8.3.2 Begrifflichkeiten und Definitionen

Folgende Begrifflichkeiten und Definitionen verwendete die Verfasserin der gegenständlichen Arbeit bei der Kodierung, Zusammenfassung und Extraktion der Erkenntnisse zu den durchgeführten Aufgaben:

Aufgabe: Bezeichnet eine thematisch klar von anderen Aktivitäten abgegrenzte Aufgabenstellung, die in weiterer Folge eindeutig durch einen Aktivitätscluster, eine Sozialform, eine Art der Aufgabe und einen Schwierigkeitsgrad identifiziert werden kann. Eine Aufgabe ist nicht weiter sinnvoll in kleinere Aufgaben zerlegbar. Eine durchgeführte Aufgabe basiert auf einem zuvor erstellten Aufgabenentwurf.

US: Kurzform für Unterstützer. Ein Unterstützer ist ein Teil des Unterstützungspersonals, das aus max. zwei Personen besteht. Hauptaufgaben des Unterstützungspersonals sind die Dokumentation des Geschehens, die Assistenz bei der Durchführung von Aufgaben und die Vorbereitung und das Wegräumen der Materialien. Zusätzlich kann ein Unterstützer bei Bedarf auch als Betreuungsperson fungieren und bei der Betreuung der Kinder unterstützen.

TR: Kurzform für Trainerin, die die jeweiligen Aufgaben erläutert und die gesamten Lehr-Lern-Einheiten leitet bzw. moderiert. Es handelt sich hierbei um die Autorin. Im Kontext der Aktivitätsschablonen des Frameworks (siehe Kapitel 8.7) ist unter TR entweder eine zum Kindergarten externe Fachperson zu verstehen, oder eine Kindergartenpädagogin bzw. ein -pädagoge.

Trockenweiterstellen: Unter dem Begriff „Trockenweiterstellen“ ist zu verstehen, dass der Bodenroboter (hier: Bee-Bot) pro eingegebenen Befehl mit der Hand um den entsprechenden Befehl weiterbewegt wird, ohne dass der Bodenroboter (hier: Bee-Bot) den

Befehl durch das Drücken der GO-Taste abarbeitet. So kann eine Befehlsabfolge visuell unterstützt eingegeben werden. In der Regel wird nach Beendigung der Befehlseingabe der Bodenroboter (hier: Bee-Bot) wieder auf seinen Ausgangspunkt zurückgestellt und durch Drücken der GO-Taste das Abarbeiten der eingegebenen Befehle gestartet.

8.3.3 Transkription und Kodierung, Inhaltsanalyse

Für die Analyse und Beurteilung der durchgeführten Aufgaben bereitete die Autorin das durch apparative Aufzeichnung bei der Durchführung der beiden Programme in den Kindergärten D und E entstandene Beobachtungsmaterial auf. Mit Hilfe der Inhaltsanalyse nach Mayring kodierte sie das Material und extrahierte anschließend die tatsächlich durchgeführten Aufgaben, um so die Grundlagen für die Analyse selbiger im Kontext eines Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zu schaffen.

Das Vorgehen untergliederte sich hierbei in folgende Teilprozessschritte:

1. Transkription der Beobachtungsvideos der zehn Lehr-Lern-Einheiten
2. Kodierung der Transkription mittels formaler Strukturierung gemäß der Inhaltsanalyse nach Mayring (siehe Kapitel 7.2.3.3). Der Leitfaden zur Kodierung ist Tabelle 3 zu entnehmen.
3. Extraktion von positiven / negativen sowie allgemeinen Erkenntnissen in Prosa-Text zu den Aufgaben gemäß Leitfaden zur Inhaltsanalyse (siehe Kapitel 8.3.4)
4. Zusammenfassen des Prosa-Texts aus Punkt 3 in positive und negative Notiz sowie Bereinigung der allgemeinen Erkenntnis. Bewertende Aussagen wurden entfernt.
5. Extraktion aller Informationen aus der allgemeinen Erkenntnis, die die eingesetzten Materialien und die Aufgabenstellung betreffen und Dokumentation unter Material und Aufgabensetting.
6. Bestimmung der tatsächlichen und der empfohlenen Dauer in Minuten
7. Definition der verwendeten Begriffe, insbesondere parallel zur Bearbeitung der Punkte 3 bis 6

Spalte	Titel	Beschreibung	Mögliche Werte
Minute	Aktuelle Minute	Hier wird die aktuelle Minute der Beobachtung verzeichnet	Die Minuten beginnen bei 0 (für Minute 0 bis Minute 1), gefolgt von 1, 2, 3 usw. bis Minute 59. Bei Minute 60 geht es mit 01:00 weiter.... Sollten für die Beobachtung einer Minute mehrere Zeilen notwendig sein, so wird erst bei der nächsten Minute wieder das Minutenfeld ausgefüllt (d.h. folgt auf eine ausgefüllte Minutenzelle eine leere, wird die Beobachtung obiger Minute zugeordnet)
Inhalt / Inhaltkürzel	Beobachteter Inhalt	Hier wird alles in Stichworten erfasst, was in der aktuellen Minute passiert ist. Einzelne Aktionen beginnen dabei jeweils in einer neuen Zeile, wobei diese zur leichteren Unterscheidung mit einem Kennbuchstaben beginnen. Hier verzeichnet werden z.B. die Aufgabenstellungen, Störungen von außen, Aktivitäten der Betreuung, der Kinder, ...	A ... Aufgabenstellung S ... Störung B ... Betreuung E ... Erklärung K ... Kind (welches im Feld Name Kind genannt wurde) KA . Alle Kinder AK . Aktivität (wenn die Kinder gerade an etwas arbeiten) AU ..Aufgabenstellung Usability
Name Kind	Name des Kindes	Name des / der aktuell im „Vordergrund“ bzw. „im Mittelpunkt“ der Beobachtung stehenden Kindes / Kinder	Alle Namen der Kinder, die an dem Tag am Programm teilgenommen haben
A Erklärung	Allgemeine Erklärung	Es wird den Kindern neuer Inhalt erklärt, dazu zählt die Erläuterung des Ablaufs ebenso, wie eine Einführung in die Thematik	0 ... nein 1 ... ja
N Aufgabe	Neue Aufgabenstellung	Den Kindern wird eine neue Aufgabenstellung gegeben	0 ... nein 1 ... ja
W Aufgabe	Wiederholung Aufgabenstellung	Wiederholung der zuletzt vorgestellten Aufgabenstellung	0 ... nein 1 ... ja
Aufmerksamkeit	Aufmerksamkeit der Kinder	Die Kinder folgen der Betreuung und stören den Ablauf nicht	0 ... alle Kinder sind nicht aufmerksam und sind nicht bei der eigentlichen Sache 1 ... $\frac{3}{4}$ der Kinder sind nicht aufmerksam 2 ... $\frac{1}{2}$ der Kinder sind nicht aufmerksam 3 ... einzelne Kinder stören 4 ... alle Kinder sind aufmerksam sind bei der Sache
Selbständigkeit	Selbständigkeit der Kinder	Die Kinder arbeiten selbständig an der Aufgabenstellung (z.B. in Gruppenaufgaben oder wenn alle gemeinsam sind).	0 ... Kinder arbeiten nicht an der Aufgabenstellung 1 ... Kinder brauchen viel Hilfe von der Betreuung 2 ... Kinder brauchen wenig Hilfe von der Betreuung

Spalte	Titel	Beschreibung	Mögliche Werte
			3 ... Kinder arbeiten selbständig ohne Hilfe der Betreuung
B Einzelaufgabe	Einzelaufgabe für ein Kind von der Betreuung	Ein Kind arbeitet im Einzelsetting an einer Aufgabenstellung, die von der Betreuung gegeben wurde	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
K Einzelaufgabe	Einzelaufgabe für ein Kind von einem Kind	Ein Kind arbeitet im Einzelsetting an einer Aufgabenstellung, die von einem anderen Kind gestellt wurde.	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
S Einzelaufgabe	Einzelaufgabe für ein Kind von dem Kind selbst	Ein Kind arbeitet im Einzelsetting an einer Aufgabenstellung, die vom Kind selbst gestellt wurde.	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
B Gruppenaufgabe	Gruppenaufgabe für alle Kinder, von der Betreuung gestellt	Alle Kinder sind gemeinsam und ein Kind nach dem anderen hat eine Aufgabe zu lösen, die von der Betreuung gestellt wurde	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
K Gruppenaufgabe	Gruppenaufgabe für alle Kinder, von einem anderen Kind gestellt	Alle Kinder sind gemeinsam und ein Kind nach dem anderen hat eine Aufgabe zu lösen, die von einem jeweils anderen Kind gestellt wurde.	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
S Gruppenaufgabe	Gruppenaufgabe für alle Kinder, jeweils vom Kind selbst gestellt	Alle Kinder sind gemeinsam und ein Kind nach dem anderen hat eine Aufgabe zu lösen, die sich das Kind jeweils selbst gestellt hat.	0 ... Kind kann die Aufgabe nicht lösen 1 ... Kind kann die Aufgabe nur mit sehr viel Unterstützung lösen 2 ... Kind kann die Aufgabe mit Hinweisen lösen 3 ... Kind kann die Aufgabe alleine lösen
H Hilfestellung	Hilfestellung	Auftreten von Hilfestellungen und Umgang mit denselben	K...von anderem Kind B...von der Betreuung

Tabelle 3: Leitfaden zur Kodierung Kindergarten D und E

8.3.4 Kategorien der Inhaltszusammenfassung

Gemäß der qualitativen Inhaltsanalyse nach Mayring (siehe Kapitel 7.2) führt die Verfasserin der gegenständlichen Arbeit nach der Strukturierung des Ausgangsmaterials (siehe Kapitel 7.2.3.3) eine Zusammenfassung (siehe Kapitel 7.2.3.1), in Hinblick auf einen möglichen Einsatz der Aufgaben in einem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ in Form von verallgemeinerten generalisierten Aktivitätsschablonen, durch.

8.3.4.1 Positive, negative und allgemeine Erkenntnisse

Für die Bewertung des Materials analysiert die Autorin positive wie auch negative Erkenntnisse bei den einzelnen Aufgaben, die durch die Transkription der Beobachtung abgeleitet werden können. Dies erfolgt in zwei Schritten: zuerst werden die positiven und negativen Erkenntnisse erfasst und kodiert. Im zweiten Schritt werden diese als positive und negative Notiz generalisiert.

- Positive Erkenntnisse: Aus der Transkription der Beobachtung der Durchführung der Aufgaben herausgefilterte positive Erkenntnisse / Begebenheiten / Erfahrungen bei der Durchführung der Aufgabe.
- Negative Erkenntnisse: Aus der Transkription der Beobachtung der Durchführung der Aufgaben herausgefilterte negative Erkenntnisse / Begebenheiten / Erfahrungen bei der Durchführung der Aufgabe.
- Positive Notiz: generalisierte positive Erkenntnisse aus der Beobachtung bei der Durchführung der Aufgabe
- Negative Notiz: generalisierte negative Erkenntnisse aus der Beobachtung bei der Durchführung der Aufgabe
- allgemeine Erkenntnis / Bemerkung zur Aufgabe: Wertfreie Beobachtungen werden in diese Kategorie kodiert und zusammengefasst.

Wenn keine Beobachtungen hinsichtlich positiver, negativer und allgemeiner Erkenntnisse aus der Transkription abgeleitet werden können, so ist das entsprechend zu vermerken.

8.3.4.2 Benötigtes Material und Aufgabensetting

Das für die jeweilige Aufgabe benötigte Material und die Beschreibung des Aufgabensettings, inkl. möglicher Fragen, die gestellt werden können und ein mögliches Vorgehen innerhalb der Aufgabe sowie allfällige Hinweise zur Durchführung werden in dieser Kategorie verzeichnet.

8.3.4.3 Tatsächliche Dauer in Minuten und Bewertung derselben

Die tatsächliche Durchführungsdauer einer Aufgabe im Lehr-Lern-Einheiten-Programm leitet die Autorin aus dem Beobachtungsmaterial ab. Aus dem Videomaterial kann die Dauer in Minuten gut abgelesen werden. Die Dauer wird von der Verfasserin der gegenständlichen Arbeit anschließend hinsichtlich der Angemessenheit, Notwendigkeit und der aufmerksamkeitsfördernden Aspekte bei dieser einen Durchführung der Aufgabe qualitativ bewertet.

Es gibt Aufgaben, für welche keine Angaben über die tatsächliche Dauer getroffen werden können. Das betrifft Aufgaben, die aus der Software-Evaluation (siehe Kapitel 9.2) abgeleitet werden können. Hier ist es auf Grund der Vielzahl an Durchführungen (pro Kind eine Durchführung) nicht möglich, eine tatsächliche Dauer zu bestimmen.

8.3.4.4 Empfohlene Dauer in Minuten

Die empfohlene Dauer der Durchführung einer Aufgabe in Minuten ist von der Autorin aus der tatsächlichen Dauer generalisiert abzuleiten.

8.3.5 Extraktion von Metadaten der Aufgaben

Zusätzlich ergänzte die Autorin aus den Beobachtungsvideos durch formale Strukturierung die Aufgaben um Metadaten, deren Leitfäden zur Auswahl der entsprechenden Attributwerte in den jeweiligen Kategorien im Folgenden näher beschrieben werden. Die Metadaten hat die Verfasserin der gegenständlichen Arbeit zum Teil deduktiv, durch theoretische Überlegungen im Vorfeld zur Kodierung des Materials definiert, zum anderen Teil induktiv aus dem Ausgangsmaterial abgeleitet. (Siehe auch Kapitel 7.2.3.1)

8.3.5.1 AufgabenID

Bei diesem Verwaltungsattribut handelt sich um eine eindeutige Identifikationsnummer einer Aufgabe um die Nachverfolgbarkeit zu gewährleisten. Die AufgabenIDs bestehen aus numerischen Werten, die fortlaufend aus vordefinierten Bereichen vergeben werden.

Aufgaben, die auf Aufgabenentwürfen für die erste Durchführung der empirischen Studie (KIGA D) basieren, erhalten eine AufgabenID kleiner als 100. Aufgaben, die auf Aufgabenentwürfen für die zweite Durchführung der empirischen Studie (KIGA E) basieren und sich (zum Teil nur in Nuancen) von KIGA D unterscheiden, erhalten eine AufgabenID zwischen 101 und 200. Weiterführende Aufgaben, die aus Aufgabenentwürfen aus der Vorstudie (siehe Kapitel 1.4) abgeleitet werden, erhalten eine AufgabenID ab 201 bis 400. Aufgaben, die auf Aufgabenentwürfen basieren und explizit der Evaluierung der Software Focus on Bee-Bot 1 bzw. 2 zugeordnet werden können, erhalten eine AufgabenID größer 400.

8.3.5.2 Sozialform

Jede Aufgabe weist eine bestimmte Sozialform auf, mit der sie durchgeführt bzw. abgehalten werden kann. Hierbei ist pro Aufgabe eine der folgenden Sozialformen zu wählen:

- Einzel in ganzer Gruppe: ein Kind bearbeitet eine Aufgabe in der Gruppe, die anderen Kinder schauen zu und unterstützen; oder ein Kind präsentiert / zeigt / erklärt der Gruppe etwas
- Einzel: jedes Kind arbeitet allein (egal ob es organisatorisch einer Kleingruppe oder der ganzen Gruppe zugeordnet ist, das wichtige ist, dass es dort allein arbeitet und die anderen Kinder gleichzeitig auch alleine arbeiten)
- Ganze Gruppe: die ganze Gruppe erarbeitet etwas gemeinsam, bespricht ein Thema, folgt einer Demonstration, etc.
- Kleingruppe in ganzer Gruppe: eine Kleingruppe bearbeitet eine Aufgabe in der Gruppe, die anderen Kinder schauen zu und unterstützen
- Kleingruppe: Kinder arbeiten in einer Kleingruppe gemeinsam

8.3.5.3 Schwierigkeitsgrad

Alle Aufgaben sind genau einem Schwierigkeitsgrad zuzuordnen. Wobei sich der Schwierigkeitsgrad an der Zielgruppe, Kindergartenkinder im Alter von 3 bis 6 Jahren orientiert. Die Skala des Schwierigkeitsgrades umfasst vier Stufen, von 1 (leicht) bis 4 (schwer).

Einflussgröße zur Einteilung der Aufgaben in ihre jeweiligen Schwierigkeitsgrade ist das voraussetzende Vorwissen der Kinder entsprechend ihrem Kindergartenbesuch in Österreich.

Für die Aufgaben sind vier Schwierigkeitsgrade mit folgenden Merkmalen definiert:

- Grad 1: Tätigkeiten, die im Kindergarten-Alltag vorkommen, als vertraut und bekannt angenommen werden dürfen. Z.B.: kleben, ausschneiden, ...
- Grad 2: Erklärungen von Technologien bzw. Geräten, Vorstellung von Funktionsdetails, einfache Aufgaben gemeinsam durch Spiel bewältigen, Gestaltung von (einfachen) Arbeitsmaterialien und zeichnen von vorgegebenen Objekten

- Grad 3: „Vorlegen“ von Wegen, die dem Roboter eingegeben werden sollen, präsentieren und erklären eigener Artefakte / Reflexionsrunden, fotografieren, Frage-Antwort-Spiel über Technik bzw. Eigenschaften von Geräten
- Grad 4: Diskussion von Fachbegriffen, „Roboter-Spielen“ – Kinder werden durch Befehlskarten / durch Befehle von anderen Kindern gesteuert, Verarbeitungsschritte eines einfachen Algorithmus erkennen, Prototyping – unbekannte Tätigkeiten

Bei Aufgaben, bei denen ein Bodenroboter programmiert wird, erfolgt eine an die Aufgaben angepasste Einteilung der Schwierigkeitsgrade, ebenfalls wieder von 1 (leicht) bis 4 (schwer) – Angaben der Merkmale aus der Sicht des Bodenroboters:

- Grad 1: nur eine Richtung, Demonstration
- Grad 2: wie Grad 1 und ein zusätzlicher Richtungswechsel, freies Erkunden
- Grad 3: wie Grad 2 und ein zusätzlicher Richtungswechsel
- Grad 4: wie Grad 3 und mindestens ein zusätzlicher Richtungswechsel, komplexere Aufgabenstellungen, Einschulung von anderen Personen in die Bedienung des Bodenroboters

Bei Aufgaben am PC bei der Nutzung einer Softwaresimulation des Bodenroboters erfolgt eine ähnliche Klassifikation der Schwierigkeitsgrade.

Die Schwierigkeitsgrade in den Übersichtsgrafiken (siehe Kapitel 8.4.3) sind wie folgt farblich kodiert:

ID w	Schwierigkeitsgrad 1
ID x	Schwierigkeitsgrad 2
ID y	Schwierigkeitsgrad 3
ID z	Schwierigkeitsgrad 4

Abbildung 12: Farbkodierung Schwierigkeitsgrade

8.3.5.4 Art der Aufgabe

Mit Art der Aufgabe wird die Tätigkeit mittels derer das Thema der Aufgabe bearbeitet wird beschrieben.

- **Bewegen:** Robotertheater, Drehungen im Stehen, nach Befehlen im Raum bewegen, nach aufgelegten Befehlskarten im Raum bewegen
- **Demonstrieren (durch Trainerin):** Trainerin zeigt der Gruppe etwas vor
- **Dokumentieren:** ein Kind erstellt ein Foto, klebt ein Foto in das Forschungsheft ein, hält Ereignisse / Dinge aus der Wirklichkeit durch Zeichnung fest, hält Datum fest, ...
- **Erforschen:** kennenlernen von Neuem, ausprobieren von Hard- und Software
- **Fragen entwickelnd unterrichten:** Trainerin erarbeitet mit der Gruppe / der Kleingruppe / einem Kind Inhalte
- **Gestalten:** ein Kind zeichnet, bastelt, beschriftet
- **Präsentieren (durch Kind):** ein Kind präsentiert der Gruppe etwas
- **Programmieren:** ein Kind oder eine Kleingruppe programmiert einen Bodenroboter oder die Simulation des Bodenroboters
- **Wiederholen:** schon Erlerntes / Besprochenes wird von der Trainerin mit den Kindern wiederholt

Die Gesamtheit der durchgeführten und extrahierten Aufgaben ist Anhang A2 zu entnehmen.

8.4 Phase 3: Konsolidierung der Aufgaben in Inhaltsdimensionen und Aktivitätscluster

Die Verfasserin der gegenständlichen Arbeit identifizierte 114 Aufgaben aus dem Beobachtungsmaterial der durchgeführten empirischen Studie und 4 Aufgaben aus der Vorstudie (die extrahierten Aufgaben sind Anhang A2 zu entnehmen). Nach erfolgter Transkription, Kodierung, inhaltlicher Analyse sowie Vervollständigung der Metadaten gilt es die Aufgaben genau einer Inhaltsdimension zuzuordnen. Die Inhaltsdimensionen entwickelte die Autorin induktiv aus der Menge der Aufgaben und den Beobachtungsergebnissen. Insgesamt liegen 118 konkrete empirisch erforschte Aufgaben vor.

8.4.1 Inhaltsdimensionen

Aus der Analyse der durchgeführten Aufgaben ermittelte die Verfasserin der gegenständlichen Arbeit induktiv aus dem ersten Drittel der Aufgaben folgende Inhaltsdimensionen, die im weiteren Verlauf der Zuordnung der übrigen Aufgaben gefestigt wurden:

- A) Programmierumgebung Hardware
- B) Informatische Lernumgebung technikfrei
- C) Wissenschaftliche Herangehensweisen
- D) Technische Alltagserfahrungen
- E) Programmierumgebung Software

Eine Inhaltsdimension repräsentiert eine Sammlung von Aufgaben, die zu einem eigenständigen Themengebiet zusammengefasst wurden. Die Anwendung der Inhaltsdimensionen als organisatorisches Element bei der Nutzung des Frameworks ist Kapitel 8.7.1 zu entnehmen. Tabelle 4 listet die Anzahl der konkreten Aufgaben pro Inhaltsdimension auf und zeigt die Anzahl der Aufgaben pro Schwierigkeitsgrad pro Inhaltsdimension.

Inhaltsdimension	Anzahl der Aufgaben				
	pro Inhaltsdimension	Schwierigkeitsgrad 1	Schwierigkeitsgrad 2	Schwierigkeitsgrad 3	Schwierigkeitsgrad 4
A) Programmierumgebung Hardware	57	9	20	13	15
B) Informatische Lernumgebung technikfrei	15	1	4	3	7
C) Wissenschaftliche Herangehensweisen	21	7	5	7	2
D) Technische Alltagserfahrungen	13	3	5	4	1
E) Programmierumgebung Software	12	3	1	3	5
Summe:	118	23	35	30	30

Tabelle 4: Anzahl der Aufgaben pro Inhaltsdimension und Schwierigkeitsgrad

8.4.2 Aktivitätscluster

Aufgaben, die thematisch zusammengehören und über ähnliche Rahmenbedingungen innerhalb einer Inhaltsdimension (siehe Kapitel 8.4.1) verfügen, sind in Aktivitätscluster zusammenzufassen. Diese Aktivitätscluster werden mit dem Großbuchstaben der

Inhaltsdimension verbunden mit einer fortlaufenden Nummer und einem Namen bezeichnet. Das Auffinden der Aktivitätscluster erfolgt induktiv. Die Anzahl der Aktivitätscluster pro Inhaltsdimension ist frei.

8.4.3 Aufgabenüberblick nach Inhaltsdimensionen und Aktivitätscluster

8.4.3.1 Inhaltsdimension A

Die Inhaltsdimension A, „Programmierungsumgebung Hardware“, beinhaltet sechs Aktivitätscluster und insgesamt 57 Aufgaben aller Schwierigkeitsgrade.

Die Aufgabenübersicht der Inhaltsdimension A ist in Abbildung 14 auf Seite 123 dargestellt. Die Ausformulierung der in der Studie verwendeten Aufgaben dieser Inhaltsdimension findet sich im Anhang A2.1.

8.4.3.2 Inhaltsdimension B

In der Inhaltsdimension B, „Informatische Lernumgebung technikfrei“ sind vier Aktivitätscluster mit insgesamt 15 Aufgaben enthalten, die überwiegend einen Schwierigkeitsgrad von 3 oder 4 aufweisen. Fünf Aufgaben sind zum Einstieg in das Thema vorhanden, diese weisen einen Schwierigkeitsgrad von 1 oder 2 auf.

Die Ausformulierung der in der Studie verwendeten Aufgaben dieser Inhaltsdimension findet sich im Anhang A2.2.

B) Informatische Lernumgebung technikfrei

B1) Verbalisierte elementare Verarbeitungsschritte	ID 113			

B2) Gestelltes Problem lösen: visualisierte Sequenzen	ID 58	ID 59		
	ID 50			

B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen	ID 53			
	ID 52			
	ID 51	ID 54	ID 55	
	ID 56	ID 57		

B4) Chaos vs. Systematik	ID 112			
	ID 49			
	ID 38			
	ID 201			

Abbildung 13: Aufgabenübersicht Inhaltsdimension B

A) Programmierumgebung Hardware

A1) Exploration Programmierumgebung	ID 119		
	ID 12	ID 17	
	ID 11		
	ID 148		

A2) Vorstellung Programmierumgebung	ID 10	ID 15	
	ID 116		

A3) Gestelltes Problem lösen: Sequenzen	ID 120	ID 121	ID 127
	ID 16	ID 122	ID 125
	ID 126		
	ID 124	ID 128	
	ID 129		

A4) Selbstgewähltes Problem lösen: Sequenzen	ID 45	ID 46	ID 140
	ID 151	ID 154	
	ID 27	ID 28	ID 29
	ID 36	ID 37	ID 39
	ID 40	ID 47	ID 133
	ID 134	ID 135	ID 136
	ID 137	ID 158	
	ID 31	ID 33	ID 41
	ID 43	ID 44	ID 152
	ID 155		
	ID 32	ID 35	ID 42
	ID 149	ID 150	ID 156
	ID 157	ID 159	ID 160

A5) Gestelltes Problem lösen: Schleifen	ID 34		

A6) Selbstgewähltes Problem lösen: Schleifen	ID 204		
	ID 202	ID 203	

Abbildung 14: Aufgabenübersicht Inhaltsdimension A

8.4.3.3 Inhaltsdimension C

Die Inhaltsdimension C, „Wissenschaftliche Herangehensweisen“, beinhaltet vier Aktivitätscluster und 21 Aufgaben.

C) Wissenschaftliche Herangehensweisen

C1) Einführung Wissensarbeit	ID 5		
	ID 7	ID 107	
	ID 8	ID 131	

C2) Forschungsheft	ID 1	ID 6	ID 9
	ID 24	ID 111	ID 143
	ID 20	ID 23	ID 142
	ID 145		

C3) Konzeption und Erstellung Lernmaterial	ID 26		
	ID 48		

C4) Festigung des Wissens und Reflexion	ID 13	ID 14	
	ID 18	ID 144	

Abbildung 15: Aufgabenübersicht Inhaltsdimension C

Die Ausformulierung der in der Studie verwendeten Aufgaben dieser Inhaltsdimension findet sich im Anhang A2.3.

8.4.3.4 Inhaltsdimension D

Die Inhaltsdimension D, „Technische Alltagserfahrungen“, ist in drei Aktivitätscluster unterteilt und verfügt über 13 Aufgaben mit allen Schwierigkeitsgraden.

D) Technische Alltagserfahrungen

D1) Technik im Alltagskontext	ID 25	ID 109	ID 110
	ID 22	ID 141	ID 161

D2) Vorwissen und Wissenstransfer	ID 2	ID 3	ID 4
	ID 102		

D3) Exploration technischer Alltagsgeräte	ID 19	ID 21	
	ID 162		

Abbildung 16: Aufgabenübersicht Inhaltsdimension D

Die Ausformulierung der in der Studie verwendeten Aufgaben dieser Inhaltsdimension findet sich im Anhang A2.4.

8.4.3.5 Inhaltsdimension E

Die Inhaltsdimension E, „Programmierungsumgebung Software“, verfügt über vier Aktivitätscluster und insgesamt 12 Aufgaben.

E) Programmierungsumgebung Software

E1) Exploration Programmierungsumgebung	ID 403	ID 409	
E2) Vorstellung Programmierungsumgebung	ID 417		
E3) Wissenstransfer	ID 401	ID 402	ID 413
	ID 405	ID 416	
E4) Selbstgewähltes Problem lösen: Sequenzen	ID 407	ID 412	
	ID 414		
	ID 418		

Abbildung 17: Aufgabenübersicht Inhaltsdimension E

Die Ausformulierung der in der Studie verwendeten Aufgaben dieser Inhaltsdimension findet sich im Anhang A2.5.

8.5 Phase 4: Entwicklung einer Tätigkeitslandkarte

Die Aufgaben sind, neben der Kategorisierung nach ihrem Inhalt, auch nach ihrer Art (siehe Kapitel 8.3.5.4) gruppierbar. In der Übersicht (siehe Tabelle 5) sind die neun Arten von Aufgaben und die jeweilige Anzahl an Aufgaben pro Schwierigkeitsgrad verzeichnet.

Art der Aufgabe	Anzahl der Aufgaben				
	pro Art der Aufgabe	Schwierigkeitsgrad 1	Schwierigkeitsgrad 2	Schwierigkeitsgrad 3	Schwierigkeitsgrad 4
Bewegen	11	1	3	1	6
Demonstrieren	11	4	3	3	1
Dokumentieren	11	6	0	4	1
Erforschen	12	1	4	3	4
Fragen entwickelnd unterrichten	7	1	0	5	1
Gestalten	5	0	3	0	2
Präsentieren	3	0	0	2	1
Programmieren	53	10	19	10	14
Wiederholen	5	0	3	2	0
Summe:	118	23	35	30	30

Tabelle 5: Anzahl der Aufgaben nach ihrer Art und Schwierigkeitsgrad

Es lässt sich folgende verallgemeinerte Tätigkeitslandkarte für das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ ableiten und als Basis für das Framework nutzen:

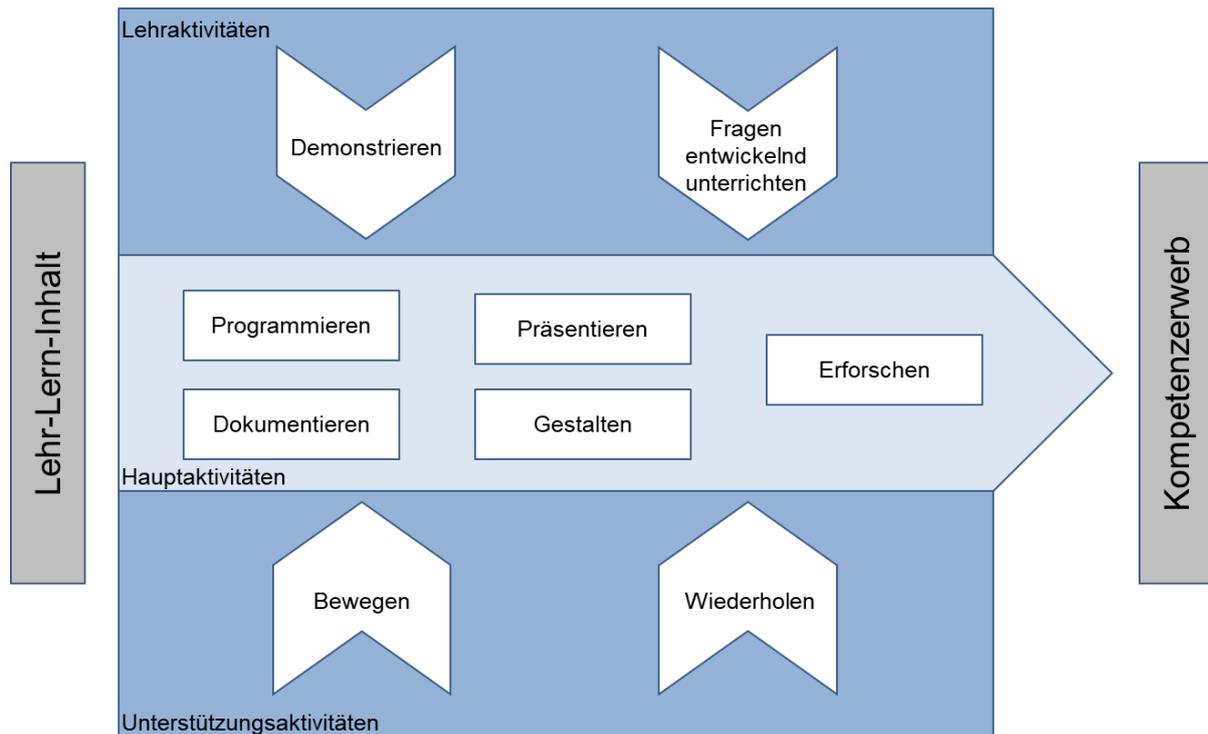


Abbildung 18: Tätigkeitslandkarte für das Framework

Die Tätigkeitslandkarte ist in die drei Bereiche Lehraktivitäten, Hauptaktivitäten und Unterstützungsaktivitäten unterteilt. Die Lehraktivitäten umfassen die Art und Weise, wie Lehr-Lern-Inhalte vermittelt werden. Die Unterstützungsaktivitäten adressieren Tätigkeiten, die die Kinder im Rahmen der Lehr-Lern-Einheiten als Unterstützung zur Erfassung von Lehr-Lern-Inhalten einsetzen können. Die Hauptaktivitäten zur Vermittlung und Erfassung der Lehr-Lern-Inhalte umfassen das Programmieren, das Dokumentieren, das Präsentieren, das Gestalten und das Erforschen.

Unter Berücksichtigung dieser Tätigkeitslandkarte können nach Prüfung der Eignung (siehe Kapitel 8.6) der konkreten Aufgaben auf die Unterstützung der zu erzielenden Lernergebnisse die verallgemeinerten Aktivitätsschablonen (siehe Kapitel 8.7) abgeleitet werden.

8.6 Phase 5: Aufgaben-Matching zu Lernergebnissen

Phase 5 der Entwicklung des Frameworks beschäftigt sich mit der Überprüfung der Aufgaben auf die zu erzielenden Lernergebnisse (siehe Kapitel 8.6.1) und der empirischen Evaluierung, ob diese Lernergebnisse tatsächlich erzielt werden können (siehe Kapitel 8.6.2).

8.6.1 Überprüfung der Eignung der Aufgaben

In diesem Schritt der Entwicklung des Frameworks werden die konkreten Aufgaben den Lernergebnissen (siehe Kapitel 8.2) zugeordnet. Hierzu wird jede konkrete Aufgabe auf ihre Lernergebnispassung überprüft (siehe Tabelle 6). Gemäß der Schwerpunktsetzung im Programm gibt es eine Häufung der Passung bei den Lernergebnissen des zweiten Leitthemas „Computational Thinking“. Die Fokussierung der Aufgaben variiert. So gibt es Aufgaben, die der Erlangung eines Lernergebnisses dienlich sind und dieses vertiefend behandeln, neben Aufgaben, deren Fokus breiter gestreut ist und die zu mehreren Lernergebnissen beitragen, dies allerdings weniger intensiv.

8.6 Phase 5: Aufgaben-Matching zu Lernergebnissen

Aufgaben ID	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5	
1																											
2							x																x				
3							x																x				
4		x					x																x			x	
5	x				x	x	x																				
6					x																						
7							x																				
8																								x			x
9					x																						
10												x	x		x	x	x							x	x		
11								x		x			x				x							x		x	
12											x	x					x	x								x	
13	x	x					x					x	x											x	x		x
14	x						x	x				x												x	x		
15								x	x	x		x						x						x	x	x	
16								x		x	x	x						x	x					x	x	x	
17								x		x	x	x	x	x				x						x	x	x	x
18							x			x																	x
19	x							x																			x
20																											x
21																											
22	x	x																									
23	x	x	x																								x
24																											
25	x																										x
26																											
27																											
28																											
29																											
31																											
32																											
33																											
34																											
35																											
36																											
37																											
38																											
39																											
40																											
41																											
42																											
43																											
44																											
45																											
46																											
47																											
48																											
49																											
50																											
51																											
52																											
53																											
54																											
55																											

8 Empirische Entwicklung eines Frameworks

Aufgaben ID	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5			
56							x			x			x	x	x			x	x	x	x				x				
57							x			x			x	x	x			x	x	x	x				x				
58												x	x					x		x						x			
59												x	x					x		x						x			
102	x						x		x								x								x	x			
107					x		x																x			x			
109	x			x																				x					
110					x																								
111					x																								
112							x								x											x			
113	x								x						x	x		x	x	x	x					x			
116							x		x			x	x		x	x	x	x						x	x				
119						x																		x	x				
120								x		x			x		x		x	x	x	x		x							
121								x		x			x		x		x	x	x	x		x							
122								x		x			x		x		x	x	x	x		x							
124								x		x			x		x		x	x	x	x		x							
125								x		x			x		x		x	x	x	x		x							
126								x		x			x		x		x	x	x	x		x							
127								x		x			x		x		x	x	x	x		x							
128								x		x			x		x		x	x	x	x		x							
129								x		x			x	x	x		x	x	x	x	x	x							
131																					x			x	x		x		
133								x		x			x	x			x	x	x		x					x			
134								x		x			x	x			x	x	x		x					x			
135								x		x			x	x			x	x	x		x					x			
136								x		x			x	x			x	x	x		x					x			
137								x		x			x	x			x	x	x		x					x			
140								x		x			x	x			x	x	x		x					x			
141	x	x																											
142			x		x																								
143					x																								
144		x			x	x																				x	x		
145					x	x	x																				x		
148						x			x			x	x								x			x			x		
149								x		x			x	x			x	x	x		x						x		
150								x		x			x	x			x	x	x		x						x		
151								x		x			x	x			x	x	x		x						x		
152								x		x			x	x			x	x	x		x						x		
154								x		x			x	x			x	x	x		x						x		
155								x		x			x	x			x	x	x		x						x		
156								x		x			x	x			x	x	x		x						x		
157								x		x			x	x			x	x	x		x						x		
158								x		x			x	x			x	x	x		x						x		
159								x		x			x	x			x	x	x		x						x		
160								x		x			x	x			x	x	x		x						x		
161	x											x																	
162					x							x													x				
201							x	x		x															x		x	x	
202							x	x		x			x						x		x						x		
203							x	x		x			x						x									x	
204						x		x	x	x																		x	
401						x	x										x	x										x	x

Aufgaben ID	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5
402							x					x	x			x	x	x	x							
403							x									x	x								x	x
405						x	x					x				x										x
407							x	x		x						x	x	x	x							
409							x									x	x									
412							x	x		x						x	x	x	x							
413							x											x							x	x
414							x	x		x						x	x	x	x							
416							x									x	x				x					
417	x								x							x	x						x			x
418							x	x		x	x		x			x	x	x	x						x	x
Σ	13	6	4	3	13	12	35	60	12	65	6	14	64	47	20	16	63	67	59	15	54	10	17	7	68	18

Tabelle 6: Zuordnung von Aufgaben zu Lernergebnissen

Wie auch aus der letzten Zeile in Tabelle 6 ersichtlich, werden durch die konkreten Aufgaben alle Lernergebnisse der Leitthemen abgedeckt.

8.6.2 Abschluss-Assessment

Zum Abschluss des Programms in den Kindergärten D und E, zeitlich nach den Hardware- und Software-Usability Tests (siehe Kapitel 9), führte die Verfasserin der gegenständlichen Arbeit Leistungstests mit den Kindern durch. Diese Leistungstests sollen Aufschluss darüber geben, welche Lernergebnisse die Kinder durch die Aktivitäten (konkrete Aufgaben, Hardware-Usability- und Software-Usability-Tests) erreicht haben. Zusätzlich arbeitete die Autorin Interviewelemente in die Leistungstests ein, um ein umfassendes Bild über die Kompetenzförderung zu erhalten.

Von den zehn aus dem Kindergarten D zum Abschluss-Assessment eingeladenen Kindern waren zum Zeitpunkt der Durchführung sieben Kinder anwesend. Diese Gruppe wird als Abschluss-Assessment-Gruppe 1 (ASG1) bezeichnet, analog zum Hardware-Usability Test und der Hardware-Usability-Test Gruppe 1 und dem Software-Usability Test und der Software-Usability-Test Gruppe 1 (siehe Kapitel 9). Hierbei handelt es sich um die Kinder, die zuerst in der Praxis den realen Bee-Bot kennengelernt haben und anschließend erst die Software Focus on Bee-Bot 1 bzw. 2.

Die Abschluss-Assessment-Gruppe 2 (ASG2) umfasst ebenfalls zehn Kinder, jedoch aus dem Kindergarten E. Bei dieser Gruppe waren zum Zeitpunkt des Tests neun Kinder anwesend. Die Kinder dieser Gruppe, auch beim Hardware- und Software-Usability Test die Testgruppe 2, lernten zuerst die Software Focus on Bee-Bot 1 bzw. 2 kennen und erst in weiterer Folge führten sie Programmierübungen am realen Bee-Bot durch.

Die Darlegung des Abschluss-Assessments beginnt die Autorin mit der Beschreibung desselbigen (Kapitel 8.6.2.1). Anschließend stellt sie die eingesetzten Methoden und Vorgehensweisen (Kapitel 8.6.2.2), die Fragestellung (Kapitel 8.6.2.3) und die zusammenfassende Transkription (Kapitel 8.6.2.4) vor. In Kapitel 8.6.2.5 legt die Verfasserin der gegenständlichen Arbeit die Entwicklung der Kategorien dar. Den Abschluss bildet sie mit der Präsentation (Kapitel 8.6.2.6) und der zusammenfassenden Interpretation der Ergebnisse (Kapitel 8.6.2.7).

8.6.2.1 Beschreibung des Assessments

8.6.2.1.1 Test-Ziel

Ziel des Abschluss-Assessments ist es herauszufinden, welche Lernergebnisse die Kinder, die an der Durchführung der konkreten Aufgaben im Rahmen der Entwicklung des

Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ teilnahmen, erreicht haben. Weiter möchte die Autorin prüfen, ob die Reihenfolge der Vorstellung von Hardware und Software eine Auswirkung auf das Ergebnis hat und ob die Kinder in der Lage sind, Aufgaben desselben Schwierigkeitsgrades wie in den Lehr-Lern-Einheiten selbständig zu lösen.

8.6.2.1.2 Kurzbeschreibung des Settings

Die Verfasserin der gegenständlichen Arbeit führte das Abschluss-Assessment im Kindergarten in einer für die Kinder vertrauten Umgebung durch, befragte die Kinder getrennt voneinander und animierte sie zur Lösung von einzelnen Aufgaben. Sowohl das Kind (K) als auch die Testleitung (TL) nahmen auf Polstern am Fußboden Platz. Vor ihnen befanden sich eine Matte mit Symbolen und ein Bodenroboter, hier ein Bee-Bot.

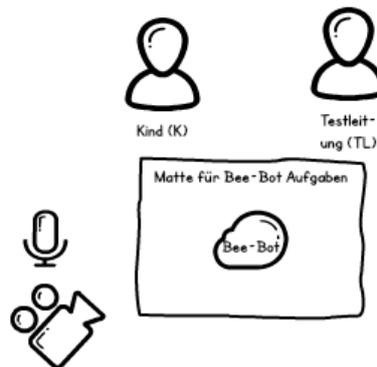


Abbildung 19: Testsetting Abschluss-Assessment

Eine Videokamera zeichnete das Abschluss-Assessment vollständig auf. Ergänzend setzte die Autorin ein Tonaufnahmegerät ein, um das gesamte Assessment zu dokumentieren und in weiterer Folge auswerten zu können.

8.6.2.1.3 Demografische Daten

Am Abschluss-Assessment nahmen insgesamt 16 Kinder teil - aus Kindergarten D zwei Mädchen und fünf Jungen, sowie aus Kindergarten E vier Mädchen und fünf Jungen.

	Weiblich	Männlich
Kindergarten D	2	5
3 Jahre	0	0
4 Jahre	0	2
5 Jahre	1	1
6 Jahre	1 ³³	2
Kindergarten E	4	5
3 Jahre	0	1
4 Jahre	1	1
5 Jahre	3	0
6 Jahre	0	3

Tabelle 7: Verteilung Alter und Geschlecht der Probandinnen und Probanden

³³ Ein Mädchen war zum Zeitpunkt des Tests noch nicht 6 Jahre alt, da sie jedoch ebenso kurz vor der Einschulung stand, wie auch die anderen 6-jährigen Kinder, wurde sie der Gruppe der Schulanfängerinnen und -anfänger im Alter von 6 Jahren zugeordnet.

8.6.2.2 Eingesetzte Methoden und Vorgehensweise

Für das Abschluss-Assessment wählte die Verfasserin der gegenständlichen Arbeit eine Kombination aus zwei Methoden für die Durchführung. Sie formulierte zu Beginn Fragen für einen sog. Leistungstest und entwickelte weiter einen Fragebogen für ein standardisiertes Interview.

Zur qualitativen Auswertung des Abschluss-Assessments griff die Autorin auf die qualitative Inhaltsanalyse nach Mayring (siehe Kapitel 7.2) zurück. Die Verfasserin der gegenständlichen Arbeit transkribierte das Assessment entlang der standardisierten Fragen pro Kind zusammenfassend und kodierte diese Zusammenfassung anschließend mittels eines Kategoriensystems, das aus den zu erwartenden Antworten und abgeleitet aus den bisherigen Erfahrungen erstellt wurde (siehe Kapitel 8.6.2.5).

Das Abschluss-Assessment fand pro Kindergarten an einem Vormittag für alle Probandinnen und Probanden hintereinander statt. In Anwesenheit der Kinder verzichtete die Testleitung weitestgehend auf das Wort „Test“, um zu vermeiden, dass für die Kinder eine Art Prüfungssituation entstehen könnte.

8.6.2.3 Fragestellung

Bei der Gestaltung der Fragestellung versuchte die Autorin nicht nur eine kindgerechte und leicht verständliche Sprache zu wählen, sondern auch die Begriffe, die in den bisherigen Aktivitäten gemeinsam mit den Kindern zum Einsatz kamen und ihnen somit vertraut sein sollten, zu verwenden. Darüber hinaus orientierte sich die Verfasserin der gegenständlichen Arbeit an den zu erzielenden Lernergebnissen (siehe Tabelle 8). Insgesamt arbeitete die Autorin 16 Fragen (und Aufgabenstellungen) aus und brachte sie im Abschluss-Assessment zum Einsatz. Naturgemäß kann im Rahmen eines Leistungstests immer nur ein Bruchteil des gesamten Lehrstoffes überprüft werden, daher orientiert sich die Fragestellung zusätzlich noch an den Forschungsfragen der gegenständlichen Arbeit. Die entwickelten Fragen für das Abschluss-Assessment lauten:

1. Was hat Dir im gesamten „Programming for Kids Kurs“ (Anmerkung: so hieß die Durchführung der Studie im Kindergarten vor Ort) am besten gefallen?
2. Was ist der Bee-Bot?
3. Wo kann man den Bee-Bot einschalten?
4. Bitte erkläre mir, wie der Bee-Bot funktioniert.
5. Der Bee-Bot soll zwei Schritte vorwärtsgehen, was musst du dafür drücken?
6. Was musst du tun, damit der Bee-Bot losfährt?
7. Der Bee-Bot soll auf der Matte, die vor uns liegt, vom Symbol Topf über das Symbol Lippe zum Symbol Fuchs fahren. Was musst du dem Bee-Bot dafür eingeben?
8. Sollte das Kind die vorherige Aufgabe mit Einzelschritten gelöst haben und nicht den Weg in einem durch eingegeben haben, dann folgende nächste Frage: Warum hast du den Bee-Bot immer in die Hand genommen und an die nächste Stelle gesetzt? Kannst du das auch ohne Einzelschritte? (Synonym zu „Trockenweiterstellen“, siehe Kapitel 8.3.2)
9. Der Bee-Bot ist in Fahrt und du siehst, dass er von der Matte herunterfährt. Wie kannst du ihn durch Tastendruck stoppen?
10. Was passiert, wenn du den Bee-Bot ausschaltest, dann wieder einschaltest und anschließend auf GO drückst? Hat sich der Bee-Bot den Weg gemerkt, den du ihm vorher programmiert hast? (Kind darf das Verhalten gerne ausprobieren)
11. Was musst du drücken, damit der Bee-Bot alles, was er gespeichert hat, löscht (vergisst)?
12. Wie viele Schritte (Befehle) kann sich der Bee-Bot merken? (Falls keine Antwort kommt, dann nachfragen: Unendlich viele Befehle?)

13. Kannst du mir sagen, was Programmieren ist?
14. Wenn du noch einmal diesen Kurs machen würdest, womit würdest du lieber Programmieren lernen. Nur mit der Software, dem Programm am Computer (Anm. d. Verf.: Focus on Bee-Bot 1 bzw. 2) oder nur mit dem Bee-Bot, den du angreifen kannst?
15. Willst du auch so Programme entwickeln, programmieren können, mit denen du am Computer mit uns gemeinsam gearbeitet hast?
16. Was möchtest du später einmal werden? (Eventuell Nachfrage zu Beruf)

Der Tabelle 8 ist die Zuordnung der Fragen des Abschluss-Assessments (AS) zu den zu überprüfenden Lernergebnissen (LE), siehe Kapitel 8.2, zu entnehmen.

Lernergebnisse	AS01	AS02	AS03	AS04	AS05	AS06	AS07	AS08	AS09	AS10	AS11	AS12	AS13	AS14	AS15	AS16
LE1.1		x	x	x												
LE1.6	x			x						x				x		
LE2.1					x	x	x	x								
LE2.2										x		x	x	x	x	
LE2.3					x		x	x								
LE2.4						x			x	x	x	x				
LE2.5			x	x						x	x	x				
LE2.6			x		x	x	x	x	x		x	x				
LE2.11							x	x								
LE3.5							x		x				x		x	x

Tabelle 8: Zuordnung Fragestellung Abschluss-Assessment (AS) zu Lernergebnissen (LE)

8.6.2.4 Zusammenfassende Transkription

Die Verfasserin der gegenständlichen Arbeit transkribiert das Videomaterial des Abschluss-Assessments und erstellt für jeden Kindergarten eine Transkriptions-Datei in Form einer Tabelle. Bei der Transkription reduziert sie den Inhalt des Ausgangsmaterials auf das Wesentliche und lässt Irrelevantes weg. Als wesentlichen Inhalt klassifiziert die Autorin, was unmittelbar als Antwort auf die Fragestellung sowohl verbal als auch nonverbal vom Kind gewertet werden kann. Da das Abschluss-Assessment auch Aufgabenstellungen beinhaltet, klassifiziert die Verfasserin der gegenständlichen Arbeit auch Aktionen des Kindes als „wesentlich“.

Die Autorin transkribiert Aussagen und Handlungen der Kinder paraphrasiert, vor allem bei direkten Fragestellungen, auf die eindeutige Antworten erkennbar sind, erfolgt eine wortwörtliche Transkription.

8.6.2.5 Kategoriensystem zur Kodierung

Die Verfasserin der gegenständlichen Arbeit führt das Abschluss-Assessment standardisiert durch (siehe Kapitel 8.6.2.2), so können die Frage- und Aufgabenstellungen direkt als Hauptkategorien übernommen werden. Die zugehörigen Unterkategorien findet die Autorin deduktiv (siehe Kapitel 7.2.3.1) aus den zu erwartenden Antworten, basierend auf den gemeinsamen Aktivitäten mit den Kindern. Daraus entsteht das Kategoriensystem für das Abschluss-Assessment in der Basisversion, Version 0.

Nach der Kodierung des ersten Kindes eines Kindergartens, das am Abschluss-Assessment teilnahm, auf Basis der Transkription der Interviews und Aufgabenstellungen sowie des Kategoriensystems in der Basisversion, nahm die Autorin eine Überarbeitung bzw. Ergänzung des Kategoriensystems vor, da weitere interessante Aspekte aus dem transkribierten Material ersichtlich wurden. Es entstand das Kategoriensystem in der Version

1. Dieses induktive Vorgehen (siehe Abbildung 20) führte zum finalen Kategoriensystem, mit dem die Verfasserin der gegenständlichen Arbeit alle 16 Teilnahmen am Abschluss-Assessment kodierte und für die Auswertung vorbereitete.

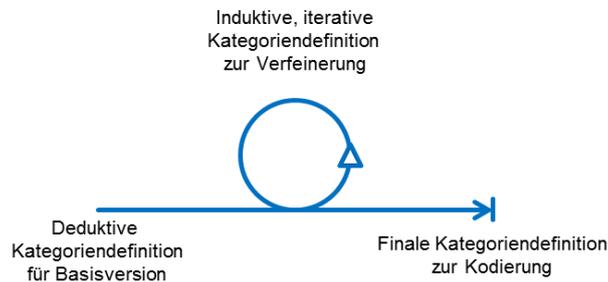


Abbildung 20: Vorgehen zur Definition des Kategoriensystems

Das finale, 99 Kategorien in 16 Hauptkategorien umfassende, und für die Kodierung des Abschluss-Assessments eingesetzte Kategoriensystem, welches die Autorin wiederum um eindeutige Identifikationsnummern für die Kategorien ergänzte, sieht wie folgt aus:

AS01	Frage 1: Am besten gefallen
AS0101	Bee-Bot (oder auch Bee-Bot auf Matte)
AS0102	Focus on Bee-Bot Software 1 bzw. 2 (oder auch am Computer)
AS0103	Putzroboter (aus den Einheiten)
AS0104	Roboter basteln oder malen
AS0105	anderes...
AS02	Frage 2: Was ist der Bee-Bot
AS0201	Ein Roboter oder auch Bodenroboter (erste Antwort)
AS0202	Eine Biene (erste Antwort)
AS0203	Ein Roboter (auf eine Nachfrage)
AS0204	anderes ...
AS03	Frage 3: Bee-Bot einschalten
AS0301	Kind schaltet von sich aus sofort und selbständig den Bee-Bot richtig ein
AS0302	Kind drückt zuerst GO
AS0303	Kind gibt an, es nicht zu wissen (erste Antwort)
AS0304	Kind schaltet auf Nachfrage(n) von TL oder Hinweis den Bee-Bot selbständig ein
AS0305	Kind kann den Bee-Bot nicht einschalten, TL schaltet Bee-Bot ein
AS04	Frage 4: Funktion Bee-Bot
AS0401	Kind drückt von sich aus sofort und selbständig Tasten am Rücken des Bee-Bots in einer sinnvollen Reihenfolge, um den Bee-Bot zum Fahren zu bringen
AS0402	Kind kann in eigenen Worten erklären, wie der Bee-Bot funktioniert
AS0403	Kind kann auf Nachfrage erklären, wie der Bee-Bot funktioniert
AS0404	Kind kann mit Hilfestellung und durch Aktionen erklären, wie der Bee-Bot funktioniert
AS0405	Kind kann nicht erklären, wie der Bee-Bot funktioniert, TL erklärt ihn durch die Hilfestellungen selbst
AS0406	Kind gibt an, es nicht zu wissen (erste Antwort)
AS05	Frage 5: Zwei Schritte vorwärts
AS0501	Kind kann von sich aus sofort und selbständig den Bee-Bot zwei Schritte vorwärts fahren lassen
AS0502	Kind zählt vorab Schritte aus
AS0503	Kind zählt während der Befehlseingabe laut mit

AS0504	Kind benötigt Hilfestellung damit der Bee-Bot fährt
AS0505	Kind benötigt Hilfestellung bei der Befehlseingabe
AS0506	Kind gibt an, es nicht zu wissen (erste Antwort)
AS0507	Kind bringt den Bee-Bot nicht zum Fahren, TL demonstriert Funktionsweise
AS0508	Kind vergisst zu Beginn die Löschen-Taste zu drücken (kann zur Irritation führen)
AS06	Frage 6: Bee-Bot losfahren lassen
AS0601	Kind antwortet "GO" (verbal oder durch zeigen)
AS0602	Kind gibt an, es nicht zu wissen (erste Antwort)
AS07	Frage 7: Komplexen Weg fahren
AS0701	Kind sitzt in Position: Blickrichtung des Bee-Bots, hinter dem Bee-Bot
AS0702	Kind sitzt nach Aufforderung in Blickrichtung des Bee-Bots
AS0703	Kind zeigt oder erklärt, bevor es mit der Programmierung beginnt, den Weg, der programmiert werden soll
AS0704	Kind erklärt während der Programmierung vor jedem Befehl, was es machen möchte
AS0705	Grad der Eigenständigkeit der Aufgabenlösung bei Einzelschritten (nachstellen des Bee-Bots)
AS0706	Anzahl der Versuche bei Einzelschritten
AS0707	Kind löst Aufgabe
AS0708	Aufgabe abgebrochen
AS0709	Kind löst Aufgabe mit kürzestem Weg
AS0710	Kind denkt von sich aus daran, vor der Eingabe der Lösung den Speicher des Bee-Bots zu löschen
AS0711	Kind denkt daran, dass nach dem Löschen der Bee-Bot wieder auf seine Ausgangsposition zurück gesetzt werden muss
AS0712	Kind hat Probleme mit links / rechts
AS0713	Kind hat Probleme mit der Art der Drehung des Bee-Bots (Bee-Bot fährt nicht links / rechts, dreht sich nur)
AS0714	Kind hat Probleme bei der Bestimmung der Anzahl von Schritten
AS0715	Kind hat Probleme bei der Eingabe der Befehle (richtige Reihenfolge bzw. richtige Anzahl)
AS08	Frage 8: Ohne Einzelschritte komplexen Weg
AS0801	Kind löst Aufgabe von sich aus, ohne Nachstellen des Bee-Bots (quasi schon bei Frage 7)
AS0802	Grad der Eigenständigkeit der Aufgabenlösung bei Gesamtprogrammierung (kein Nachstellen)
AS0803	Anzahl der Versuche bei Gesamtprogrammierung
AS0804	Kind löst Aufgabe
AS0805	Aufgabe abgebrochen
AS0806	Kind löst Aufgabe mit kürzestem Weg
AS0807	Kind denkt von sich aus daran, vor der Eingabe der Lösung den Speicher des Bee-Bots zu löschen
AS0808	Kind denkt daran, dass nach dem Löschen der Bee-Bot wieder auf seine Ausgangsposition zurück gesetzt werden muss
AS0809	Kind hat Probleme mit links / rechts
AS0810	Kind hat Probleme bei der Bestimmung der Anzahl von Schritten
AS0811	Kind hat Probleme bei der Eingabe der Befehle (richtige Reihenfolge bzw. richtige Anzahl bzw. vertippt)
AS0812	Kind gibt an, Einzelschritte verwendet zu haben, weil es einfacher ist
AS0813	Kind gibt an, die Aufgabenstellung auch ohne Nachstellen (Einzelschritte) lösen zu können
AS09	Frage 9: Bee-Bot stoppen
AS0901	Kind drückt auf GO bzw. antwortet "GO" bzw. antwortet "grün" oder zeigt auf "GO"

AS0902	Kind drückt auf Clear-Taste bzw. zeigt auf Clear-Taste bzw. antwortet "vergessen" o.ä.
AS0903	Kind gibt an, es nicht zu wissen (erste Antwort)
AS10	Frage 10: Ausschalten, Einschalten, GO
AS1001	Kind antwortet sofort: "Nein, die Biene hat alles vergessen" o.ä.
AS1002	Kind gibt an, dass die Biene nichts vergessen hat bzw. dass keine Befehle gelöscht wurden
AS1003	Kind gibt an, es nicht zu wissen (erste Antwort)
AS1004	Nach Ausführung der drei Tasten-Betätigungen ist Kind der Überzeugung, dass Befehle noch gespeichert sind
AS1005	Kind benötigt mehr Hilfestellungen (als das Durchführen der drei Aktionen), um zur Antwort zu kommen (dass Befehle nicht mehr gespeichert sind)
AS11	Frage 11: Löschen
AS1101	Kind antwortet "Vergessen" und zeigt auf Clear-Taste o.ä.
AS1102	Kind gibt an, es nicht zu wissen (erste Antwort)
AS12	Frage 12: Anzahl der Befehle, die gespeichert werden können
AS1201	Kind nennt 40 Befehle (erste Antwort)
AS1202	Kind nennt eine Zahl kleiner als 40 (erste Antwort)
AS1203	Kind nennt eine Zahl größer als 40 aber kleiner, gleich als 100 (erste Antwort)
AS1204	Kind nennt eine Zahl größer als 100 (erste Antwort)
AS1205	Kind nennt unendlich / beliebig viele (erste Antwort)
AS1206	Kind bejaht auf Nachfrage und Hinweisen eine Zahl rund um 40
AS1207	Kind gibt an, es nicht zu wissen (erste Antwort)
AS1208	Kind bekräftigt auch auf Nachfrage, dass es nicht weiß, wie viele Schritte gespeichert werden können
AS1209	Kind gibt keine Antwort
AS13	Frage 13: Programmieren erklären
AS1301	Kind bringt von sich aus Programmieren mit Befehlseingabe am Bee-Bot in Verbindung
AS1302	Kind gibt an, es nicht zu wissen (erste Antwort)
AS1303	Kind nennt Speichern als Programmieren (erste Antwort)
AS1304	Kind nennt Knöpfe zum Einschalten als Programmieren (erste Antwort)
AS1305	Kind nennt das manuelle Bewegen des Bee-Bots als Programmieren und vergisst auf die notwendige Befehlseingabe
AS14	Frage 14: Präferenzen
AS1401	Kind bevorzugt Bee-Bot
AS1402	Kind bevorzugt Software Focus on Bee-Bot 1 bzw. 2
AS1403	Kind gibt an, es nicht zu wissen (erste Antwort) und Antwort erfolgt auf Nachfrage bzw. Hilfestellung
AS1404	Kind hat keine Präferenzen
AS1405	Grund: es gefällt mir
AS1406	Grund: weil es eine Spielzeug-Biene ist
AS1407	Grund: Kind gefällt die Schwierigkeit bzw. die Möglichkeit zur Steigerung der Schwierigkeit
AS1408	Grund: Kind erkennt, dass das Wissen vom Bee-Bot auf die Software übertragen werden kann
AS15	Frage 15: Programme entwickeln
AS1501	Kind möchte Spiele bzw. Programme entwickeln
AS1502	Kind gibt an, es nicht zu wissen (erste Antwort)
AS16	Frage 16: Berufswunsch
AS1601	Kind möchte einen technischen Beruf ergreifen

AS1602	Kind möchte einen wissenschaftlichen Beruf ergreifen
AS1603	Kind möchte einen sozialen / medizinischen Beruf ergreifen
AS1604	Kind möchte einen anderen Beruf ergreifen
AS1605	Kind hat keinen konkreten Berufswunsch
AS1606	Kind gibt an, es nicht zu wissen (erste Antwort)

Tabelle 9: Kategoriensystem Abschluss-Assessment, finale Version

Die Hinweise zur Kodierung des Kategoriensystems sind in Anhang A3 zu finden.

8.6.2.6 Ergebnisse

Die Ergebnisse des Abschluss-Assessments werden entlang der Kategorien quantitativ ausgewertet und in einer zusammenfassenden Interpretation (siehe Kapitel 8.6.2.7) qualitativ betrachtet.

8.6.2.6.1 *AS01 Frage 1: Am besten gefallen*

Ein Kind von 7 Kindern (14 %) der Gruppe ASG1 gibt an, dass ihm der Hardware Bee-Bot von allen Aktivitäten, die im Rahmen des Programms „Programming for Kids“ durchgeführt wurden, am besten gefallen hat. In der Gruppe ASG2 sind es 2 der 9 Kinder (22 %).

In der Gruppe ASG1 geben 2 der 7 Kinder (29 %) an, dass ihnen die Aktivitäten mit der Software Focus on Bee-Bot 1 bzw. am besten gefallen haben, wohingegen in der Gruppe ASG2 kein Kind diese als Präferenz nennt.

Großen Anklang fand die Beschäftigung mit dem Putzroboter, die von 4 Kindern der Gruppe ASG2 als beliebteste Aktivität genannt wurde und von einem Kind der Gruppe ASG1. Je 2 Kinder pro Gruppe nannten die Bastel- und Zeichenaktivitäten rund um Roboter als ihre liebsten Aktionen. Je ein Kind pro Gruppe nannte andere Tätigkeiten, die ihm am besten gefallen haben.

8.6.2.6.2 *AS02 Frage 2: Was ist der Bee-Bot*

Auf die Frage, was der Bee-Bot sei, antwortete die Mehrheit der befragten Kinder, dass es sich um eine Biene handelt. 5 der 7 Kinder (71 %) der Gruppe ASG1 identifizierten in ihrer jeweiligen ersten Antwort den Bee-Bot als Biene (3 davon sogar als einen Bienenroboter). 2 Kinder identifizierten zuerst eine Biene und gaben auf Nachfrage an, dass es sich um einen Roboter handelt. 2 Kinder gaben gleich bei der ersten Antwort an, dass es sich beim Bee-Bot um einen Roboter handelt.

In der Gruppe ASG2 wurde diese Frage nur 8 Kindern gestellt. 6 der 8 Kinder (75 %) identifizierten den Bee-Bot als eine Biene, ein Kind als einen Bienen-Roboter. 3 Kinder gaben auf Nachfrage an, dass es sich um einen Roboter handelt und 2 Kinder gaben als erste Antwort an, dass es sich um einen Roboter handelt. Ein Kind war der Meinung, dass es sich um keinen Roboter handelt.

8.6.2.6.3 *AS03 Frage 3: Bee-Bot einschalten*

In der Gruppe ASG1 konnten alle Kinder den Bee-Bot auf Anhieb selbständig und ohne Hilfe einschalten. In der Gruppe ASG2 wurde 8 Kindern die Aufgabe des Einschaltens gestellt und davon konnten 6 Kinder den Bee-Bot auf Anhieb selbständig und ohne Hilfe einschalten. Ein Kind versuchte den Bee-Bot mittels GO-Taste einzuschalten und ein Kind gab an, nicht zu wissen wie dieser einzuschalten ist. Mit Hilfe der Testleitung konnten alle Kinder den Bee-Bot einschalten.

8.6.2.6.4 AS04 Frage 4: Funktion Bee-Bot

Die Funktionalität des Bee-Bots demonstrierten 4 der 7 Kinder (57 %) in der Gruppe ASG1 und 3 der 9 Kinder (33 %) in der Gruppe ASG2 auf Anhieb am Bee-Bot. Von allen befragten Kindern konnte ein Kind aus der Gruppe ASG2 in eigenen Worten erläutern, wie der Bee-Bot funktioniert. 6 der 7 Kinder (86 %) der Gruppe ASG1 und ein Kind von 9 Kindern (11 %) der Gruppe ASG2 konnten auf Nachfrage die Funktionalität erklären. Wobei in der Gruppe ASG1 3 Kinder sowohl durch Demonstration als auch durch Sprache die Funktionalität erklärten.

4 der 9 Kinder in der Gruppe ASG2 konnten mit Unterstützung der Testleitung die Funktionalität erklären. Weiter konnte in der Gruppe ASG2 ein Kind die Funktionalität gar nicht erklären und ein Kind gab an, nicht zu wissen wie der Bee-Bot funktioniert, konnte die Funktionalität dann aber mit Hilfestellung erklären.

8.6.2.6.5 AS05 Frage 5: Zwei Schritte vorwärts

In der Gruppe ASG1 konnten 6 der 7 Kinder (86 %) den Bee-Bot auf Anhieb und ohne Hilfestellung zwei Schritte vorwärts fahren lassen. 2 Kinder zählten vor der Programmierung die Schritte aus und nur ein Kind zählte laut bei der Eingabe der Befehle mit. Ein Kind benötigte bei der Befehlseingabe Hilfestellung.

In der Gruppe ASG2 konnten 4 der 9 Kinder (44 %) den Bee-Bot auf Anhieb und ohne Hilfestellung zwei Schritte vorwärts fahren lassen. 7 Kinder zählten vorab die Schritte laut aus und 2 Kinder zählten während der Befehlseingabe laut mit. 3 Kinder benötigten Hilfestellung, damit der Bee-Bot fährt, 4 Kinder benötigten Hilfestellung bei der Befehlseingabe. Davon benötigten 2 Kinder sowohl bei der Befehlseingabe als auch beim „zum Fahren bringen“ Hilfestellung. 5 Kinder vergaßen zu Beginn der Programmierung mit Betätigen der Clear-Taste den Speicher zu löschen, um zu verhindern, dass allfällig gespeicherte Befehle zu Beginn ausgeführt werden.

8.6.2.6.6 AS06 Frage 6: Bee-Bot losfahren lassen

Alle Kinder konnten den Bee-Bot losfahren lassen und somit die GO-Taste sinngemäß bedienen.

8.6.2.6.7 AS07 Frage 7: Komplexen Weg fahren

4 der 7 Kinder in der Gruppe ASG1 bearbeiteten die komplexe Aufgabenstellung mit Einzelschritten (davon erklärte, zeigte ein Kind vor der Programmierung den Lösungsweg und ein Kind erklärte vor jedem Programmierschritt, was es programmieren möchte), 3 Kinder bearbeiteten von sich aus die Aufgabenstellung ohne Einzelschritte (davon erklärten, zeigten 2 Kinder vor der Programmierung den Lösungsweg).

3 der 4 Kinder, die die Aufgabe mit Einzelschritten („Trockenweiterstellen“) bearbeiteten, lösten die Aufgabe und nutzten einen kürzesten Weg. Bei einem Kind wurde die Aufgabenlösung nach vier Versuchen abgebrochen, um das Kind nicht zu demotivieren. Dieses Kind hatte Probleme bei der Bestimmung der Anzahl der Schritte und bei der Eingabe der Befehle, auch in Hinblick auf die Reihenfolge der Eingabe von Befehlen.

4 der 7 Kinder der Gruppe ASG1 saßen bei der Aufgabenbearbeitung hinter dem Bee-Bot, mit gleicher Blickrichtung wie der Bee-Bot, ein Kind saß nach Aufforderung durch die Testleitung in entsprechender Position. 3 der Kinder nutzten Einzelschritte („Trockenweiterstellen“) zur Bearbeitung der Aufgabe und 2 der Kinder versuchten es ohne.

In der Gruppe ASG2 bearbeiteten alle 9 Kinder die Aufgabenstellung mit Hilfe von Einzelschritten („Trockenweiterstellen“). Alle Kinder saßen in der Sitzposition hinter dem Bee-Bot und mit gleicher Blickrichtung wie der Bee-Bot, ein Kind erst nach Aufforderung. 3 der Kinder erklärten vorab den Weg, den sie programmieren möchten und 3 andere Kinder erklärten vor jedem Programmierbefehl, was sie als nächstes tun werden.

4 der 9 Kinder lösten die Aufgabe, bei 5 Kindern wurde die Aufgabe abgebrochen, nachdem es nicht möglich erschien, dass das Kind die Aufgabe lösen kann bzw. die Möglichkeit bestand, dass das Kind demotiviert und enttäuscht reagiert.

4 Kinder setzten bei der Bearbeitung an, den kürzesten Weg zu wählen – 3 lösten die Aufgabe tatsächlich mit dem kürzesten Weg. Ein Kind der 9 Kinder dachte daran, vorab den Speicher zu löschen und 3 Kinder dachten daran, dass sie den Bee-Bot, bei Löschen des Speichers, wieder in seine Ausgangsposition für die Aufgabenstellung zurückstellen müssen. 3 der Kinder, die die Aufgabe nicht gelöst haben, hatten Probleme mit links / rechts, ein weiteres Kind hatte Probleme mit dem Verständnis der Art und Weise, wie der Bee-Bot sich dreht (und dass er bei einer Drehung keinen Schritt macht). 4 der 5 Kinder, die die Aufgabe nicht gelöst haben, hatten Probleme mit der Bestimmung der Anzahl der Schritte, die der Bee-Bot zurücklegen muss, ein weiteres Kind hatte dieses Problem ebenso, konnte aber die Aufgabe dennoch lösen. 7 der 9 Kinder der Gruppe ASG2 hatten Probleme bei der Eingabe von Befehlen, sei es bei der Reihenfolge oder der richtigen Anzahl.

8.6.2.6.8 AS08 Frage 8: Ohne Einzelschritte komplexen Weg

Die Lösung der komplexen Programmieraufgabe ohne Einzelschritte probierte in der Gruppe ASG2 nur ein Kind und brach diese nach neun Versuchen ab. Das Kind hatte die Aufgabe zuvor mit Hilfe von Einzelschritten gelöst („Trockenweiterstellen“) und gab auch an, dass ihm dies wesentlich leichter fiel, als alle Befehle auf einmal einprogrammieren zu müssen, denn gerade die Anzahl an einzugebenden Befehlen stellte hier eine Herausforderung dar.

In der Gruppe ASG1 versuchten 6 Kinder, die Aufgabe ohne Einzelschritte zu lösen, 2 sogar von sich aus, ohne vorher die Aufgabe mit Einzelschritten („Trockenweiterstellen“) zu probieren und ein Kind wechselte gleich zu Beginn zur Aufgabenbearbeitung ohne Einzelschritte. Der Grad der Eigenständigkeit bei der Aufgabenbearbeitung war hoch und die Anzahl der Versuche schwankte zwischen einem und vier, wobei meistens ein Versuch nötig war. 5 der 6 Kinder lösten die Aufgabe und nahmen dabei einen der möglichen kürzesten Wege. 3 Kinder hatten Probleme bei der Eingabe der Befehle, ein Kind hatte ob der Komplexität der Aufgabenstellung Probleme mit links / rechts. Das Löschen von eventuell früher eingespeicherten Befehlen stellte keine Probleme dar.

3 der Kinder, die auf die Frage antworteten, ob die Lösung der Aufgabe mit Einzelschritten („Trockenweiterstellen“) leichter sei, beantworteten diese mit ja.

8.6.2.6.9 AS09 Frage 9: Bee-Bot stoppen

5 von 6 befragten Kindern in der Gruppe ASG1 nutzten die GO-Taste zum Stoppen des Bee-Bots, ein Kind wollte den Bee-Bot durch die Clear-Taste zum Stehen bringen. In der Gruppe ASG2 verwendeten 5 der 7 Kinder die GO-Taste zum Stehen bleiben des Bee-Bots, 2 Kinder die Clear-Taste.

8.6.2.6.10 AS10 Frage 10: Ausschalten, Einschalten, GO

Die Fragestellung, ob der Bee-Bot seine Befehle im Speicher hält, auch wenn er ausgeschaltet wird, wurde von den Kindern differenziert beantwortet. So meinten 4 der 6 befragten Kinder (66 %) in der Gruppe ASG1, dass der Speicher geleert wird und 2 Kinder waren der Meinung, dass die Befehle im Speicher erhalten bleiben. Ein Kind war zuerst der Überzeugung, dass die Befehle im Speicher erhalten bleiben und konnte dann mit Unterstützung durch die Testleitung nachvollziehen, dass der Speicher beim Ausschalten gelöscht wird.

In der Gruppe ASG2 waren 2 der 7 Kinder (29 %) der Meinung, dass der Speicher gelöscht wird und 5 Kinder (71 %) gaben an, dass die Befehle im Speicher erhalten bleiben. Auch in dieser Gruppe gab es ein Kind, das zu Beginn der Überzeugung war, dass die Befehle erhalten bleiben und durch gemeinsame Erarbeitung mit der Testleitung zum Schluss kam, dass dem doch nicht so ist.

8.6.2.6.11AS11 Frage 11: Löschen

Alle Kinder wussten auf Anhieb, dass mit der Taste „Clear“ alle gespeicherten Befehle gelöscht werden.

8.6.2.6.12AS12 Frage 12: Anzahl der Befehle, die gespeichert werden können

Kein Kind gab an, dass der Bee-Bot bis zu 40 Befehle speichern kann. 2 Kinder meinten bei ihrer ersten Antwort, dass der Bee-Bot weniger als 40 Befehle speichern kann, 2 Kinder meinten zwischen 40 und 100 Befehle. 4 Kinder gaben an, dass der Bee-Bot unendlich viele Befehle speichern kann, 4 Kinder kamen auf Nachfragen zu einer Zahl von ca. 40 Befehlen. 5 Kinder gaben bei ihrer ersten Antwort an, dass sie es nicht wissen und ein Kind bekräftigte sogar, trotz Nachfragen und Hinweisen, dass es das nicht weiß. Ein Kind enthielt sich der Antwort.

8.6.2.6.13AS13 Frage 13: Programmieren erklären

2 der 7 Kinder (29 %) in der Gruppe ASG1 und 3 der 9 Kinder (33 %) in der Gruppe ASG2 konnten den Begriff „Programmieren“ erklären bzw. mit den Aktivitäten rund um den Bee-Bot in Verbindung bringen. Die Hälfte der Kinder konnte bei ihrer ersten Antwort nicht erklären, was Programmieren ist. Ein Kind brachte das Speichern von Befehlen am Bee-Bot mit Programmieren in Zusammenhang, ein Kind meinte, dass das Einschalten des Bee-Bots das Programmieren ist. Ein Kind gab keine Antwort.

8.6.2.6.14AS14 Frage 14: Präferenzen

Der Bee-Bot gefiel 10 der 16 Kinder (63 %) am besten, gefolgt von der Software Focus on Bee-Bot 1 bzw. 2, die 6 der 16 Kinder (37 %) bevorzugten, wenn es darum ging, mit welchen der beiden Materialien Kinder beginnen sollten, in die Welt der Programmierung einzutauchen. Ein Kind gab als Grund für das Gefallen des Bee-Bots das bloße Gefallen an, ein Kind bevorzugte den Bee-Bot weil er eine Spielzeug-Biene ist. 2 Kinder bevorzugten die Software, weil dort der Schwierigkeitsgrad variieren kann und der sich vermutlich noch steigern könnte. 2 Kinder gaben als Grund für das Gefallen des Bee-Bots bzw. der Software an, dass es ihnen gefällt, dass sie das Wissen von dem einen auf das andere übertragen können und dass beides ähnlich zu bedienen ist.

8.6.2.6.15AS15 Frage 15: Programme entwickeln

10 der 13 befragten Kinder gaben an, dass sie selbst einmal ein Computerspiel oder ein Programm für einen Computer entwickeln und programmieren möchten.

8.6.2.6.16AS16 Frage 16: Berufswunsch

Einen technischen Beruf wollten zum Zeitpunkt des Abschluss-Assessments 5 der 16 Kinder ergreifen. Einen wissenschaftlichen Beruf haben 3 Kinder in Erwägung gezogen, 2 Kinder zeigten Interesse an einem sozialen oder medizinischen Beruf. 6 Kinder wollten einen Beruf einer anderen Berufsgruppe ergreifen.

8.6.2.7 Zusammenfassende Interpretation der Ergebnisse und Schlussfolgerungen

Rund 20% der Kinder, die am Abschluss-Assessment teilnahmen, empfanden die Beschäftigung mit dem Hardware Bee-Bot als beste von allen Aktivitäten (AS01), die im Rahmen des durchgeführten Programms angeboten und abgehalten wurden. Die überwiegende Mehrheit der Kinder erkannte im Bee-Bot eine Biene (AS02), jedoch nur ca. ein Drittel der befragten Kinder identifizierte in ihm einen Roboter.

Das Einschalten des Bee-Bots (AS03) stellte für die Kinder keine allzu große Hürde dar. Die meisten Kinder konnten den Bee-Bot auf Anhieb selbständig und ohne Hilfe einschalten, mit ein wenig Hilfe durch die Testleitung konnten alle Kinder den Bee-Bot einschalten. Interessant zu beobachten ist, dass das Kind, welches zum Einschalten die GO-Taste

betätigte, diese auch als erstes beim Hardware-Usability-Test betätigte und dann aber den Bee-Bot selbständig einschalten konnte.

Bei der Fragestellung nach der Funktionalität des Bee-Bots (AS04) zeigt sich, dass es 3- bis 6-jährigen Kindern nicht leichtfällt, ihr Wissen so darzulegen, dass auch andere dieses Wissen nutzen können. Einige Kinder demonstrieren von sich aus die Funktionalität und können dadurch zeigen, dass sie die Funktionalität des Bee-Bots kennen. Wenn die Kinder in ein Gespräch über die Funktionalität verwickelt werden, dann gelingt es ihnen durchaus, die Funktionalität zu erklären.

Alle Kinder konnten den Bee-Bot zwei Schritte vorwärts (AS05) fahren lassen, etwa die Hälfte der Kinder zählte zuerst die zu fahrenden Schritte aus (Vorausplanen) und 3 Kinder zählten laut mit, während sie die Befehle in den Bee-Bot eingaben. Somit zeigt sich, dass Kinder, nach Besuch des Programms, in der Lage sind, einfache Programmieraufgaben mit dem Bee-Bot selbständig zu lösen.

Die Lösung komplexer Aufgaben (AS07) mit dem Bee-Bot ist für manche Kinder nicht mehr ganz so einfach. 7 der 13 Kinder (54 %), die die Aufgabe unter Einsatz von Einzelschritten („Trockenweiterstellen“) bearbeiteten, lösten die Aufgabe tatsächlich. Dabei schwankte der Grad der Eigenständigkeit (die Aufgabe ohne Hilfestellung zu lösen) zwischen 3 (keine Hilfestellung) und 0 (sehr viel Hilfestellung). Die meisten Probleme hatten die Kinder bei der Eingabe der richtigen Befehlsreihenfolge bzw. Anzahl der Schritte (8 der 13 Kinder), bei der Bestimmung der zu fahrenden Anzahl an Schritten (6 der 13 Kinder) und der Unterscheidung zwischen links und rechts (3 der 13 Kinder).

Bei der Bearbeitung der komplexen Aufgabe ohne Einsatz von Einzelschritten (AS08) („Trockenweiterstellen“), ist zwar die Erfolgsrate deutlich höher, 5 der 7 Kinder (71 %) lösten die Aufgabe, doch es haben zum einen weniger Kinder probiert und die 4 Kinder, die auf die Frage antworteten, ob die Bearbeitung mit Einzelschritten („Trockenweiterstellen“) leichter sei, bejahten diese. Für die Kinder scheint die Lösung einer komplexen Aufgabenstellung ohne Einzelschritte deutlich herausfordernder zu sein, da nicht nur vorausgeplant, sondern auch eine größere Befehlsanzahl auf einmal eingegeben werden muss. Bei der Verwendung von Einzelschritten hingegen wird nach jeder einzelnen Befehlseingabe der jeweilige Schritt evaluiert und dann wiederum ein Befehl eingegeben.

Es zeigt sich hier, dass 3- bis 6-jährige Kinder nach Besuch des Programms in der Lage sind komplexe Wege bzw. komplexe Befehlsabfolgen mit dem Bee-Bot zu bestreiten. Weiter zeigt sich, dass ein Einsatz des realen Bee-Bots zu Beginn des Programms, gefolgt von einem Software-Einsatz, positive Auswirkungen auf das Gesamtergebnis hat. Die Kinder sind eher in der Lage komplexe Aufgabenstellungen zu lösen, sogar ohne der Methode „Trockenweiterstellen“ (siehe Kapitel 8.3.2). Die umgekehrte Reihenfolge, bei der zuerst die Software und später die Hardware zum Einsatz kommen, ist bedingt empfehlenswert. Die Lernerfolge sind im Vergleich zurückhaltender.

Tendenziell fällt Kindern im Alter von 5 oder 6 Jahren die Programmierung etwas leichter als jüngeren, aber auch 3- und 4-Jährige sind in der Lage, den realen Bee-Bot sinngemäß zu nutzen und Wege korrekt zu programmieren, wenn auch noch nicht ganz so komplexe.

Die Mehrheit der Kinder (10 der 13 Kinder, 77 %) wusste, dass der fahrende Bee-Bot durch ein erneutes Drücken der GO-Taste zum Stehen gebracht werden kann (AS09). 3 Kinder wollten dies durch Betätigung der Clear-Taste erwirken.

Betrachtet man die Fragestellung, ob der Bee-Bot nach dem Ausschalten und wieder Einschalten alle Befehle gelöscht hat (AS10), über alle Kinder, die an dem Abschluss-Assessment teilgenommen haben, so sind die Antworten ausgeglichen: 6 der 13 befragten Kinder waren überzeugt, dass der Bee-Bot keine Befehle mehr gespeichert hat, 7 Kinder hingegen waren überzeugt, dass der Bee-Bot noch Befehle gespeichert hat. Sehr große Unterschiede gab es jedoch in den zwei Gruppen. Nur 2 Kinder waren nach dem Ausprobieren von Ausschalten, Einschalten und GO-Taste betätigen noch überzeugt, dass

Befehle eingespeichert waren – sie änderten aber ihre Meinung nach entsprechenden Erklärungen und Demonstrationen durch die Testleitung.

Das bewusste Löschen aller gespeicherten Befehle (AS11) im Bee-Bot stellte kein Kind vor eine Herausforderung. Alle Kinder waren in der Lage durch Nennung oder Betätigung der Taste „Clear“, den Speicher des Bee-Bots zu leeren.

Die Anzahl an möglichen Befehlen (AS12), die der Bee-Bot speichern kann dürfte für die Kinder, auf Grund der Zahlengröße, etwas Abstraktes sein, das sie sich schwer vorstellen können. Kein Kind gab an, dass der Bee-Bot bis zu 40 Befehle speichern kann, 4 Kinder kamen auf eine ähnliche Zahl mit Nachfragen und Hilfestellungen. 4 Kinder meinten sogar, dass der Bee-Bot unendlich viele Befehle speichern kann – das kann vermutlich daran liegen, dass in den Lehr-Lern-Einheiten des Programms kaum die Speicherkapazität ausgereizt wurde und die Kinder daher keine Gelegenheit hatten zu erfahren, was es bedeutet, dass der Speicher voll ist.

Das Erklären des Begriffs „Programmieren“ (AS13) stellt für die Kinder eine Herausforderung dar. Zum einen fällt ihnen das Erklären in eigenen Worten schwer, was die Autorin auch bei den früheren Tests feststellte, zum anderen ist Programmieren für Kinder ein sehr abstrakter Begriff, der nicht leicht nachzuvollziehen ist. Fasst man die Kategorien AS1301, AS1303, AS1304 und AS1305 unter dem Argument zusammen, dass dies alles zumindest im Ansatz mit der Eingabe von Befehlen oder dem Herbeiführen eines Ergebnisses zu tun hat, dann hätten 9 der 16 Kinder, also knapp mehr als die Hälfte, zumindest versucht, den Begriff im Kontext des Bee-Bots zu erklären.

Die Kinder bevorzugen zum ersten Kennenlernen von Programmierung und für die ersten Programmieraktivitäten (AS14) den realen Bee-Bot gegenüber der Software Focus on Bee-Bot 1 bzw. 2. Neben Begründungen, die mit dem bloßen Gefallen des Bee-Bots zu tun haben, ist es bemerkenswert, dass es 2 Kindern ein Bedürfnis war der Testleitung mitzuteilen, dass sie es toll finden, das Wissen vom realen Bee-Bot auf die Software übertragen zu können und vice versa. Das spricht, neben obigen Ergebnissen, dafür, dass es sich empfiehlt unbedingt einen kombinierten Ansatz mit dem Einsatz von Hardware und Software zu wählen. Es ist jedoch gleichermaßen wichtig, zuerst mit der Hardware zu beginnen, da dies die Vorstellungskraft der Kinder stärkt und die Abstraktion zu Beginn etwas mindert, so dass Kindern der Einstieg in die Programmierung leichter fällt. Der Einsatz der Software erscheint vor allem auch deswegen wichtig, weil 2 Kinder explizit darauf hinwiesen, dass bei dieser der Schwierigkeitsgrad der Aufgaben stärker variieren und daher auch die Spannung und Neugierde länger aufrechterhalten werden kann.

Nach den abgehaltenen Lehr-Lern-Einheiten waren 10 der 13 befragten Kinder der Meinung, dass sie selbst einmal ein Computerspiel oder ein Programm entwickeln oder programmieren möchten (AS15). Auch wenn man das wohl als 3- bis 6-jähriges Kind noch nicht wirklich sagen kann, ob einen das später auch einmal interessieren wird, so ist das doch ein sehr interessantes Ergebnis: Die Kinder haben keine Ablehnung bez. Programmieren und haben ein erstes Gefühl dafür entwickeln können.

8.6.2.7.1 Überprüfung der Lernergebnisse

Die Überprüfung, ob die Lernergebnisse erzielt wurden, erfolgt durch eine qualitative Konsolidierung der Beantwortung der einzelnen Aufgaben und Fragstellungen des Abschluss-Assessments.

Nach Besuch der Lehr-Lern-Einheiten sind die Kinder in der Lage, ...

Lernergebnis 1.1: ... technische Begriffe in eigenen Worten zu erklären.

Kindern ist es kein Leichtes über ihr Wissen mit anderen zu sprechen. Es fällt ihnen leichter, Dinge und Vorgänge durch Vorzeigen zu vermitteln. Vor allem abstrakte Elemente, wie z.B. der Bee-Bot, der in der Form einer Biene einen Roboter tarnt, stellen hinsichtlich verbaler Beschreibungen eine Herausforderung dar. Die Aufgaben tragen dazu bei, die Kinder zu

ermutigen über ihnen nicht so vertraute Gegenstände, Begriffe und Vorgänge zu sprechen und so die Geläufigkeit hierzu zu trainieren.

Lernergebnis 1.6: ... über einen längeren Zeitraum Erlebtes und Erlerntes zusammenfassend wieder zu geben.

Es ist den Kindern möglich, über Erlebtes und Erlerntes, welches sie über einen längeren Zeitraum erworben haben, zu reflektieren und dieses in eigenen Worten wiederzugegeben. Besonders das Sprechen über persönliche Präferenzen fällt ihnen leicht. Die im Zuge des Programms zur frühen Technikbildung in Informatik durchgeführten konkreten Aufgaben bieten Möglichkeit zur Reflexion und unterstützen die Kinder bei einer zusammenfassenden Wiederholung des Erlebten und Erlernten.

Lernergebnis 2.1: ... einfache und komplexe Programmier-Probleme zu lösen.

Alle Kinder konnten den Bee-Bot zwei Schritte vorwärts fahren lassen, etwa die Hälfte der Kinder zählte zuerst die zu fahrenden Schritte aus (vorausplanen). Es gelingt Kindern nach Besuch der Lehr-Lern-Einheiten einfache Programmieraufgaben selbständig mit dem Bodenroboter, hier dem Bee-Bot, zu lösen. Ebenso ist es ihnen möglich, den Bodenroboter sinngemäß zu nutzen. Bei komplexen Aufgaben ist erhebliche Hilfestellung notwendig, um einen geeigneten Lösungsweg für das Programmierproblem zu ermitteln. Hier hilft vor allem die Zerlegung der komplexen Problemstellungen in kleinere Aufgaben bzw. Einzelschritte. Die Aufgaben zeigen den Kindern auf, dass die Zerlegung größerer Probleme in kleinere Teilprobleme hilfreich sein kann und fördern so die selbständige Lösungsfindung komplexerer Programmier-Probleme.

Lernergebnis 2.2: ... informatische Begriffe in eigenen Worten zu erklären.

Hier zeigt sich ein ähnliches Bild wie bei Lernergebnis 1.1. Gerade abstrakte Begriffe, wie z.B. Programmieren, stellen Kinder vor erhebliche Herausforderungen hinsichtlich einer Erklärung des Begriffs. Wobei weniger die Tätigkeit an sich ein Problem darzustellen scheint, als die korrekte Verwendung der Vokabel und die Verknüpfung des eigenen Wissens mit dieser für sie zum Teil abstrakten Welt. Ein regelmäßiges Üben und Auseinandersetzen mit für Kinder neuartigen Begriffen, wie eben z.B. Programmieren, unterstützt die Kinder dabei, in eigenen Worten auch komplexe und abstrakte Vorgänge sowie Begriffe zu beschreiben und Dritten zu erklären.

Lernergebnis 2.3: ... einen komplexen Weg in Einzelschritte zu zerlegen.

Die Analyse zeigt, dass Kindern das Finden von Lösungen für komplexe Programmieraufgaben wesentlich leichter fällt, wenn sie zuerst die Hardware des Bodenroboters kennengelernt haben und dann die Softwaresimulation des Bodenroboters. Die Kinder zerlegen zur Lösungsfindung den komplexen Weg meistens in Einzelbefehle (Schritte des (Boden)roboters) – dazu wird das sog. Trockenweiterstellen (siehe Kapitel 8.3.2) genutzt. In der Softwaresimulation kann dies über den Befehl „Einzelschritte anzeigen“ realisiert werden. Eine direkte Eingabe der Befehle zur Lösung, ohne dass Einzelschritte des virtuellen (Boden)roboters in der Softwaresimulation zwischenzeitlich durch Befehlsausführung angezeigt werden, stellt Kinder vor eine zum Teil nicht unerhebliche Herausforderung. Hier sind adäquate Hilfestellungen und regelmäßiges Üben ähnlicher Aufgabenstellungen angebracht.

Lernergebnis 2.4: ... Ursache-Wirkungs-Prinzipien zu erkennen.

Kinder sind in der Lage Ursache-Wirkungs-Prinzipien zu erkennen und die Aufgaben tragen dazu bei, das Identifizieren selbiger zu fördern. Durch Ausprobieren möglicher Tastenkombinationen können Kinder Rückschlüsse auf das Verhalten des Roboters ziehen und erfahren, welche „Wirkung“ so manche „Ursache“ haben kann.

Lernergebnis 2.5: ... die grundlegende Funktionalität eines Bodenroboters, der über eine Mini-Language verfügt, zum Beispiel anhand des Bee-Bots, zu erklären.

Die Erläuterung der Funktionalität eines Bodenroboters, der über eine Mini-Language verfügt, bereitet den Kindern große Probleme. Nur ein Kind konnte beim Abschluss-Assessment auf Anhieb in eigenen Worten die Funktionalität des Bodenroboters erläutern.

Mit Hilfestellungen bzw. Nachfragen konnten zahlreiche weitere Kinder die Funktionalität erklären. Mit Hilfe von Demonstrationen und Vorzeigen konnten Kinder sich wesentlich leichter ausdrücken und so ihr Wissen präsentieren. Es zeigt sich, wie auch schon bei Lernergebnis 1.1 bzw. 2.2, dass hier regelmäßiges üben und eine adäquate Unterstützung notwendig sind, um den Kindern die Möglichkeit zu bieten, für sie neue und abstrakte Begriffe in eigenen Worten inhaltlich korrekt zu erklären.

Lernergebnis 2.6: ... einen Bodenroboter zur Lösung von Aufgabenstellungen sinngemäß zu bedienen.

Die grundlegende sinngemäße Bedienung des Bodenroboters zur Lösung einer einfachen Aufgabenstellung, vom Einschalten, über das Eingeben von Befehlen und das Starten der Ausführung selbiger bis zum Löschen der eingegebenen Befehle und Eingeben einer neuen Befehlsabfolge gelingt den Kindern nach Besuch der Lehr-Lern-Einheiten gut. Bei der Planung und Eingabe komplexerer Programmieraufgaben, greifen die Kinder auf die Problemlösungsmethode „Trockenweiterstellen“ zurück und zerlegen so komplexe Probleme in kleine einfache Einzelprobleme. Auch hier sind eine sinngemäße Verwendung bzw. Bedienung des Bodenroboters klar erkennbar. Wenn die Kinder aufgefordert werden, eine andere Problemlösungsmethode als „Trockenweiterstellen“ einzusetzen, dann stellt dies eine sehr große Herausforderung dar und intensive Hilfestellungen und Betreuungen der Kinder beim Lösen der Aufgaben sind notwendig – wenn sich die Kinder überhaupt solchen Herausforderungen stellen möchten. Üben und Wiederholen fördern das Ausbilden unterschiedlicher Problemlösungsstrategien.

Lernergebnis 2.11: ... sich im Raum zu orientieren, die Richtungen zu benennen und anzuwenden.

Durch Besuch der Lehr-Lern-Einheiten entwickeln die Kinder Strategien um sich effizienter im Raum zu orientieren und Richtungsanweisungen korrekt zu erkennen bzw. programmieren zu können. Besonders der Einsatz von Befehlskärtchen zur Visualisierung der zu programmierenden Befehle bzw. des Speicherinhalts und von Bewegungsaufgaben, bei denen Kinder die Positionen des Bodenroboters einnehmen und am eigenen Körper die Auswirkungen von Dreh-Befehlen, Schritten vorwärts bzw. rückwärts erleben, helfen den Kindern hierbei. Auch die Übungen im Sitzkreis, wo die Kinder Einzelaufgaben mit dem Bodenroboter auf einer Matte lösen sollen tragen dazu bei, dass die Kinder, gerade auch durch das Feedback von den anderen Kindern, erfahren, wie sie sich im Raum orientieren können („setz dich immer hinter den Bodenroboter, dann ist es leichter“, „links ist dort wo du dein Armband hast“). Schwieriger wird es für Kinder, besonders bei Entscheidungen zwischen Links und Rechts, wenn sie sich nicht mit Hilfestellungen wie z.B. dem Trockenweiterstellen behelfen und versuchen, komplexe Richtungsabfolgen im Kopf zu entwickeln und zu programmieren. Hier ist das Betreuungspersonal gefordert, adäquate Hilfestellungen zu geben und oben genannte Methoden zur Problemlösung (z.B. Befehlskärtchen) einzusetzen, um die Kinder dabei zu unterstützen, korrekte Richtungsanweisungen zu geben.

Lernergebnis 3.5: ... das eigene Handeln zu kommentieren.

Das Kommentieren des eigenen Handelns fällt Kindern wesentlich leichter, als das Erklären von Fachbegriffen. Besonders beim Programmieren sind Kinder leicht zu animieren, vorab darüber zu sprechen, welchen Weg sie planen zu programmieren. Sie sprechen darüber, welche Befehle sie als nächstes eingeben möchten und begründen stellenweise ihre Entscheidungen. Auf Nachfrage finden die meisten Kinder eine plausible Erklärung für ihr Handeln. Kinder sind nach Besuch der Lehr-Lern-Einheiten in der Lage auch Präferenzen zu ihnen neuen Themen kundzutun und die Auswahl zu begründen. Kinder können in eigenen Worten erzählen, warum sie z.B. Zeichnungen derart gestaltet haben, wie sie vorliegen.

8.6.2.7.2 Exemplarische Unterschiede zwischen ASG1 und ASG2

Die Ergebnisse aus der Kategorie AS01 (siehe Kapitel 8.6.2.6.1) sind im Kontext der Reihenfolge des Kennenlernens von Software und Hardware sehr interessant. Die Kinder,

die zuerst die Software kennengelernt hatten (ASG2), bevorzugten den Putzroboter (Hardware) und kein Kind dieser Gruppe die Software als Lieblingsaktivität. Alle bis auf ein Kind bevorzugten, wenn man Putzroboter und Bee-Bot unter Hardware subsummiert, Hardware. Kein Kind bevorzugte Software.

Hingegen präferierten die Kinder der Gruppe ASG1, die zuerst die Hardware und dann die Software kennenlernte, mit 2 Kindern zu einem Kind die Software gegenüber der Hardware. Zählt man den Putzroboter noch zur Hardware hinzu, dann sind die Vorlieben zwischen Hardware und Software gleich gelagert.

Es scheint, dass, wenn Kinder mit der Hardware beginnen Programmieren zu lernen und anschließend mit der Software arbeiten, das Interesse bzw. die Bevorzugung der Materialien in der Zielgruppe annähernd gleich verteilt ist. Erfolgt der Einstieg in die Programmierung jedoch mit der Software und kommt die Hardware erst später zum Einsatz, scheint die Software uninteressant zu werden und die Aktivitäten mit Hardware gewinnen absolut an Bedeutung bzw. steigt das Interesse an diesen deutlich.

Bei der Beschreibung des Bee-Bots (AS02) differiert diese wenig. Bis auf ein Kind sind sich alle Kinder einig, dass es sich beim Bee-Bot um einen Roboter handelt und fast 70 % der befragten Kinder erkannten im Bee-Bot zusätzlich eine Biene. Nur ein Kind der Gruppe ASG2 erkannte lediglich eine Biene im Bee-Bot und sprach das Wort Roboter nicht aus. Für die Definition, worum es sich beim Bee-Bot handelt, macht es sichtlich keinen Unterschied, ob mit der Software oder mit der Hardware begonnen wird, Programmieren zu lernen.

Die Erläuterung der Funktionalität (AS04) fiel den Kindern der Gruppe ASG1 deutlich leichter, als den Kindern der Gruppe ASG2. Das dürfte daran liegen, dass die Kinder der Gruppe ASG1 den Bee-Bot zuerst kennenlernten, ihr Wissen bei der Nutzung der Software festigen konnten und in weiterer Folge die Fertigkeiten in der Bedienung der Hardware weiter ausbauen konnten. Den Kindern der Gruppe ASG1 fiel es leichter als den Kindern der Gruppe ASG2 durch Demonstration die Funktionalität zu zeigen oder auf Nachfrage die Funktionalität zu erörtern.

Damit der Bee-Bot zwei Schritte vorwärts fährt (AS05), muss zu Beginn sichergestellt werden, dass der Speicher leer ist (Clear-Taste betätigen), dann wird zweimal die „Vorwärts“-Taste betätigt, gefolgt von der GO-Taste. Das gelang 6 der 7 Kinder (86 %) der Gruppe ASG1 und nur eines der Kinder benötigte Hilfestellung bei der Befehlseingabe. Nur 2 Kinder zählten vorab die Anzahl der zu fahrenden Schritte aus und davon zählte ein Kind auch die Anzahl der eingegebenen Befehle mit.

In der Gruppe ASG2 brachten nur 4 der 9 Kinder den Bee-Bot auf Antrieb zum Fahren von zwei Vorwärts-Schritten. Jedoch zählten hier deutlich mehr Kinder vor der Programmierung die Anzahl der zu fahrenden Schritte aus (7 der 9 Kinder, 78 %). Davon zählten 2 Kinder bei der Eingabe der Befehle laut mit. 5 Kinder benötigten Hilfestellung bei der Eingabe von Befehlen bzw. damit der Bee-Bot tatsächlich losfährt. 5 Kinder vergaßen zu Beginn den Speicher zu löschen bzw. waren durch die Ausführung von nicht unmittelbar mit der Aufgabenstellung eingegebenen Befehlen durch den Bee-Bots bei Druck auf die Taste „GO“ irritiert.

Die Gruppe ASG2 hatte die Lehr-Lern-Einheiten mit der Nutzung der Software begonnen und es scheint, dass es 3- bis 6-jährigen Kindern schwerer fallen dürfte, das in einer Software erworbene Wissen auf entsprechende Hardware anzuwenden.

Bei der Programmierung komplexer Wege (AS07) zeigte sich, dass die Gruppe ASG1 deutlich besser abschnitt als die Gruppe ASG2. Nicht nur, dass mehr Kinder gleich vorneweg die Aufgabe ohne Einzelschritte bearbeiteten, sondern auch der Grad der Eigenständigkeit war deutlich höher und die Anzahl der Versuche geringer. Die meisten Probleme zeigten sich bei der Gruppe ASG2, wo vor allem die korrekte Eingabe der Befehle in den Bee-Bot für die Kinder zur Herausforderung wurde.

Auch beim Einsatz einer Gesamtprogrammierung, ohne Einzelschritte zu verwenden (AS08), schnitt die Gruppe ASG1 deutlich besser ab, als die Gruppe ASG2. In der Gruppe ASG2 probierte lediglich ein Kind die Programmierung ohne Einzelschritte und der Test wurde nach neun Versuchen beendet. In der Gruppe ASG1 versuchten 6 Kinder die Aufgabe ohne Einzelschritte zu lösen, 2 davon ohne zuvor die Aufgabe mit Einzelschritten („Trockenweiterstellen“) gelöst zu haben und 5 der 6 Kinder waren erfolgreich. Der Grad der Eigenständigkeit ist auch hier sehr hoch, wobei natürlich die Anzahl der Hilfestellungen im direkten Vergleich zur Lösung mit Einzelschritten zugenommen hat.

Während es beim Stoppen des Bee-Bots (AS09) keine erkennbaren Unterschiede zwischen den beiden Gruppen gab, zeigen sich diese wiederum stark, wenn es darum geht, ob der Bee-Bot nach dem Ausschalten noch Befehle gespeichert hat (AS10) und diese nach dem Einschalten durch Betätigung der Taste „GO“ abgerufen werden können. 4 der 6 befragten Kinder (66 %) in der Gruppe ASG1 waren überzeugt, dass der Bee-Bot die Befehle nach dem Ausschalten vergisst, 2 waren der gegenteiligen Meinung. In der Gruppe ASG2 waren jedoch 2 der 7 befragten Kinder (29 %) der Meinung, dass der Bee-Bot alle Befehle nach dem Ausschalten gelöscht hat. Aber 71 % (5 der 7 Kinder) meinten, dass der Bee-Bot nach Ausschalten und erneutem Einschalten noch immer die vor dem Ausschalten eingegebenen Befehle gespeichert hat.

Hier kann vermutet werden, dass die Kinder der Gruppe ASG1, die den Bee-Bot vor der Software kennengelernt haben, vertrauter waren mit der Tatsache, dass die Hardware die Befehle nur solange der Bee-Bot eingeschaltet ist speichert. Die Kinder der Gruppe ASG2 lernten die Programmierung des Bee-Bots zu Beginn mittels Software und stiegen erst später auf den Bee-Bot um. In der Software wird der Bee-Bot weder ein- noch ausgeschaltet und der Speicher wird nur auf Aufforderung des Kindes gelöscht. Dies könnte zur Fehlinterpretation des tatsächlichen Verhaltens des realen Bee-Bots führen.

Alle Kinder wussten, wie der Speicher des Bee-Bots bewusst gelöscht (AS11) werden kann. Jedoch bei der Anzahl an speicherbaren Befehlen (AS12) waren sich die Kinder nicht mehr einig. Während sowohl in der Gruppe ASG1, als auch in der Gruppe ASG2 jeweils ein Kind eine Anzahl kleiner als 40 und jeweils ein Kind eine Anzahl größer 40 und kleiner als 100 vermuteten, sind in der Gruppe ASG1 3 Kinder der Meinung, dass der Bee-Bot unendlich viele Befehle speichern kann. In der Gruppe ASG2 ist nur ein Kind dieser Meinung. Dies beruht wahrscheinlich auf der Tatsache, dass die Kinder in der Gruppe ASG2 bei der ersten Verwendung der Software durch ihre Experimente und ihr Ausprobieren viel eher in die Lage kamen, dass die maximale Anzahl an Befehlen für den virtuellen Bee-Bot gespeichert wurde, als das bei den Kindern in der Gruppe ASG1 beim realen Bee-Bot zu Beginn der Fall war.

Während die Kinder der Gruppe ASG1 unmittelbar gesehen haben, welche(n) Weg(strecke) diese Anzahl an Befehlen mit sich bringt, konnten die Kinder in der Gruppe ASG2 zu Beginn der Nutzung der Software nur vermuten. Sie erkannten aber recht bald, dass der Speicher schnell gefüllt werden kann und dieser nicht unbegrenzt ist. Die Kinder in der Gruppe ASG1 kamen bei ihren ersten Versuchen mit dem realen Bee-Bot kaum in die Lage, 40 Befehle einzugeben und zu erkennen, dass der Bee-Bot nicht mehr speichern kann.

Das Erklären des Begriffs „Programmieren“ (AS13) fällt den Kindern der beiden Gruppen annähernd gleich schwer. Pro Gruppe gab rund ein Drittel der Kinder eine adäquate Antwort und in der Gruppe ASG1 setzten 2 weitere Kinder noch die Betätigung von Tasten bzw. das Speichern am Bee-Bot mit Programmieren gleich. In der Gruppe ASG2 versuchte ein Kind mittels Demonstration zu erklären, was Programmieren ist und bewegte dazu auch den Bee-Bot. Das Kind vergaß dabei aber auf die Befehlseingabe am Rücken des Bee-Bots und damit auf den wichtigen Teil, der zum gewünschten Verhalten des Bee-Bots führt.

Die Kinder in beiden Gruppen bevorzugten gleich stark den Bee-Bot vor der Software Focus on Bee-Bot 1 bzw. 2 als Einstieg in die Welt der Programmierung (AS14). Auch in beiden Gruppen ist der Anteil an Kindern, die der Meinung sind, dass sie später gerne einmal ein

Computerspiel oder ein Programm für einen Computer entwickeln möchten, annähernd gleich groß. Es scheint, dass die Entscheidung, ob mit der Hardware oder mit der Software begonnen wird, programmieren zu lernen, keinen Einfluss darauf hat, ob die Kinder Interesse an einer späteren Eigenentwicklung von Software zeigen. Die Kinder sind sich mit zwei Drittel zu einem Drittel der Stimmen einig, dass der Einstieg in die Programmierung mit dem realen Bee-Bot zu bevorzugen ist.

8.6.2.7.3 Exemplarische geschlechterspezifische Unterschiede

Die Interpretation der Daten in Bezug auf geschlechterspezifische Unterschiede ist mit Vorsicht zu betrachten. Mit dem vorhandenen Datenmaterial können keine verallgemeinerbaren Aussagen getroffen werden, sondern lediglich die Beobachtungen im Rahmen der stattgefundenen Aktivitäten widerspiegelt werden.

83 % der Mädchen konnten die Aufgabe, den Bee-Bot zwei Schritte vorwärts fahren zu lassen, auf Anhieb und selbständig lösen, während dies bei den Jungen 50 % auf Anhieb schafften (AS0501). 4 von 5 Mädchen, die die Aufgabe auf Anhieb lösten, zählten die zu fahrenden Schritte vorab aus – sie planten den Weg voraus, während das kein Junge, der die Aufgabe auf Anhieb löste, machte. Interessanterweise planten die Jungen, die die Aufgabe nicht auf Anhieb gelöst hatten, vorab den Weg und zählten die Schritte aus (50 %). Sie konnten in weiterer Folge die Aufgabe mit Hilfestellungen lösen (AS0502, AS0503, AS0504, AS0505). Es scheint, dass die Mädchen bedachter an die Aufgabe herangehen und durch Vorausplanung versuchen, die Aufgabe zu lösen. Bei Jungen zeigt sich, entweder sie können es auf Anhieb, ohne weitere Umwege, oder aber sie brauchen Unterstützung und nutzen Hilfsmittel um zur Lösung zu kommen. Also quasi die beiden Extreme, aber nicht der Mittelweg, den man durchaus nutzen könnte. Das gilt aber vorerst nur für einfache Aufgaben.

Bei den komplexen Aufgaben (AS07 und AS08) zeigt sich, dass in Anlehnung an die einfachen Aufgaben, Mädchen eher mit Einzelschritten („Trockenweiterstellen“) versuchen die Aufgaben zu lösen und Jungen hier nicht mehr so eindeutig wie bei den einfachen Aufgaben agieren. 100 % der Mädchen versuchten die Aufgabe mit Einzelschritten zu lösen. 83 % der Mädchen lösten die komplexe Aufgabe mit Einzelschritten, sie benötigten dazu zwischen einem Versuch und vier Versuchen. Davon erklärten 60 % der Mädchen vor der Befehlseingabe, welchen Schritt sie als nächsten tun werden und planten den Schritt voraus, 20 % der Mädchen planten den gesamten Weg voraus und programmierten anschließend die Einzelschritte.

Bei den Jungen versuchten 70 % die Aufgabe mit Einzelschritten („Trockenweiterstellen“) zu lösen, 30 % gingen gleich in eine Lösung ohne Einzelschritte. Von den 70 % lösten 29 % der Jungen die Aufgabe erfolgreich und benötigten dazu zwischen einem Versuch und zwei Versuchen. 100 % der Jungen, die die Aufgabe mit Einzelschritten lösten, planten und erklärten vorab den Weg, kein Junge plante während der Aufgabenlösung einen Weg neu bzw. erklärte die nächsten Schritte.

Es zeigt sich also in Summe, dass die Mädchen durch ihre Vorausplanung, das laute Mitreden und Auszählen der nächsten Schritte eher zu einem guten Ergebnis kommen als die Jungen. Auch bei der Befehlseingabe erscheinen die Mädchen eigenständiger (AS0705) als die Jungen. Die größten Probleme haben aber beide Gruppen bei der Bestimmung der Anzahl der Schritte (AS0714) und bei der Eingabe der richtigen Reihenfolge der Befehle (AS0715).

Tendenziell taten sich die Jungen bei der Lösung der komplexen Aufgabe ohne Einzelschritte leichter (AS08), jedoch ist die Grundmenge deutlich zu klein, um hierüber eine klare Aussage treffen zu können. Vor allem da bei knapp der Hälfte der Kinder, die am Abschluss-Assessment teilnahmen, nach der Übung mit den Einzelschritten („Trockenweiterstellen“) der Eindruck entstand, dass die Lösung der Aufgabe ohne Einzelschritte die Kinder vor ein für sie unlösbares Problem stellen würde. Daher wurde auf

die Ausführung dieser Aufgabe bei ihnen verzichtet. Hierzu waren keine geschlechtsspezifischen Unterschiede beobachtbar.

Weiter interessant zu beobachten war, dass alle Mädchen den Bee-Bot korrekt stoppen konnten, wohingegen bei den Jungen nur knapp die Hälfte die GO-Taste zum Stoppen einsetzten (AS0901). Bei der Frage, ob der Bee-Bot den Speicher beim Aus- und wieder Einschalten löscht, halten sich die Ja- und Nein-Antworten bei Mädchen und Jungen annähernd die Waage (AS1001).

Ebenso zu erwähnen ist die Tatsache, dass alle Mädchen einstimmig den realen Bee-Bot als erstes Material zum Erlernen von Programmierung vorschlugen, sich jedoch nur 40 % der Jungen für diese Reihenfolge aussprachen. 60 % der Jungen schlugen zu Beginn den Einsatz der Software vor und erst später einen Einsatz des realen Bee-Bots. Die Mehrheit der Jungen, die den Einsatz der Software vorgeschlagen hatten, löste die komplexe Aufgabe (AS08) ohne Einsatz von Einzelschritten eher eigenständig und erschien daher eher vom realen Bee-Bot unterfordert zu sein.

Zusammengefasst scheint der reale Bee-Bot, unabhängig vom Geschlecht, für Anfängerinnen und Anfänger in der Programmierung geeignet und mit zunehmender Erfahrung der Kinder ist ein Umstieg auf die Software anzudenken. Mädchen versuchen ihre Aufgaben wohlüberlegt zu lösen und reden auch gerne über ihre Lösungsansätze. Jungen verbergen eher ihre Lösungsideen, planen etwas weniger vor und versuchen Aufgaben unüberlegter dafür schneller zu lösen. Es zeigt sich, gerade auch bei diesem Abschluss-Assessment, dass Mädchen und Jungen unterschiedliche Lösungsstrategien einsetzen, jedoch Mädchen mit ihren Strategien eher Erfolg haben als Jungen. Ähnliches ist der Literatur zu entnehmen, wie in Kapitel 2.5.1 dargestellt.

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Im letzten Entwicklungsschritt der Erstellung des Frameworks leitete die Autorin aus den konkreten Aufgaben (siehe Kapitel 8.4.3) verallgemeinerte, generalisierte Aktivitätsschablonen ab.

Für die verallgemeinerten Aktivitätsschablonen generierte die Autorin eine vierstellige ID, die sich an der ersten Stelle aus dem Identifikator der Inhaltsdimensionen (siehe Kapitel 8.4.1), gefolgt von der Nummer der Aktivitätscluster (siehe Kapitel 8.4.2) und vom Schwierigkeitsgrad der Aufgabe (siehe Kapitel 8.3.5.3), in Form von „S“ und dem Schwierigkeitsgrad (Zahl von 1 bis 4), zusammensetzt.

Bei der Ausformulierung der Aktivitätsschablonen berücksichtigte die Verfasserin des Frameworks die Kriterien zur zielgruppengerechten Gestaltung von Aufgaben im Kontext der Informatik von Dagiene & Futschek (2008) (siehe Kapitel 4.2).

Bei einer zukünftigen Ableitung von konkreten Aufgaben aus dem Framework und bei der Durchführung selbiger ist die Theorie von Vygotskys „Zone of proximal development“ (siehe Kapitel 2.1.5) zu berücksichtigen. Ebenso gilt es darauf zu achten, dass, wie in Kapitel 2.5.4 dargelegt, für die Kinder gleichgeschlechtliche Mentorinnen und Mentoren bei der Ausführung der aus dem Framework abgeleiteten Aufgaben zur Verfügung stehen.

Das finale, abstrahierte, allgemeingültige Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ besteht aus 50 Aktivitätsschablonen, die den drei Leitthemen in fünf Inhaltsdimensionen und zu neun Arten von Tätigkeiten sowie vier Schwierigkeitsgraden zugeordnet sind. Abbildung 21 zeigt einen schematischen Überblick über die vier Ordnungsdimensionen des Frameworks.

Auf Basis dieser Ordnungsdimensionen kann das Framework aus vier unterschiedlichen Perspektiven betrachtet und eingesetzt werden:

- Framework Perspektive 1: Aktivitätsschablonen nach Inhaltsdimensionen (siehe Kapitel 8.7.1)
- Framework Perspektive 2: Aktivitätsschablonen nach Leitthemen (siehe Kapitel 8.7.2)
- Framework Perspektive 3: Aktivitätsschablonen nach Lernergebnissen (siehe Kapitel 8.7.3)
- Framework Perspektive 4: Aktivitätsschablonen nach Tätigkeiten (siehe Kapitel 8.7.4)

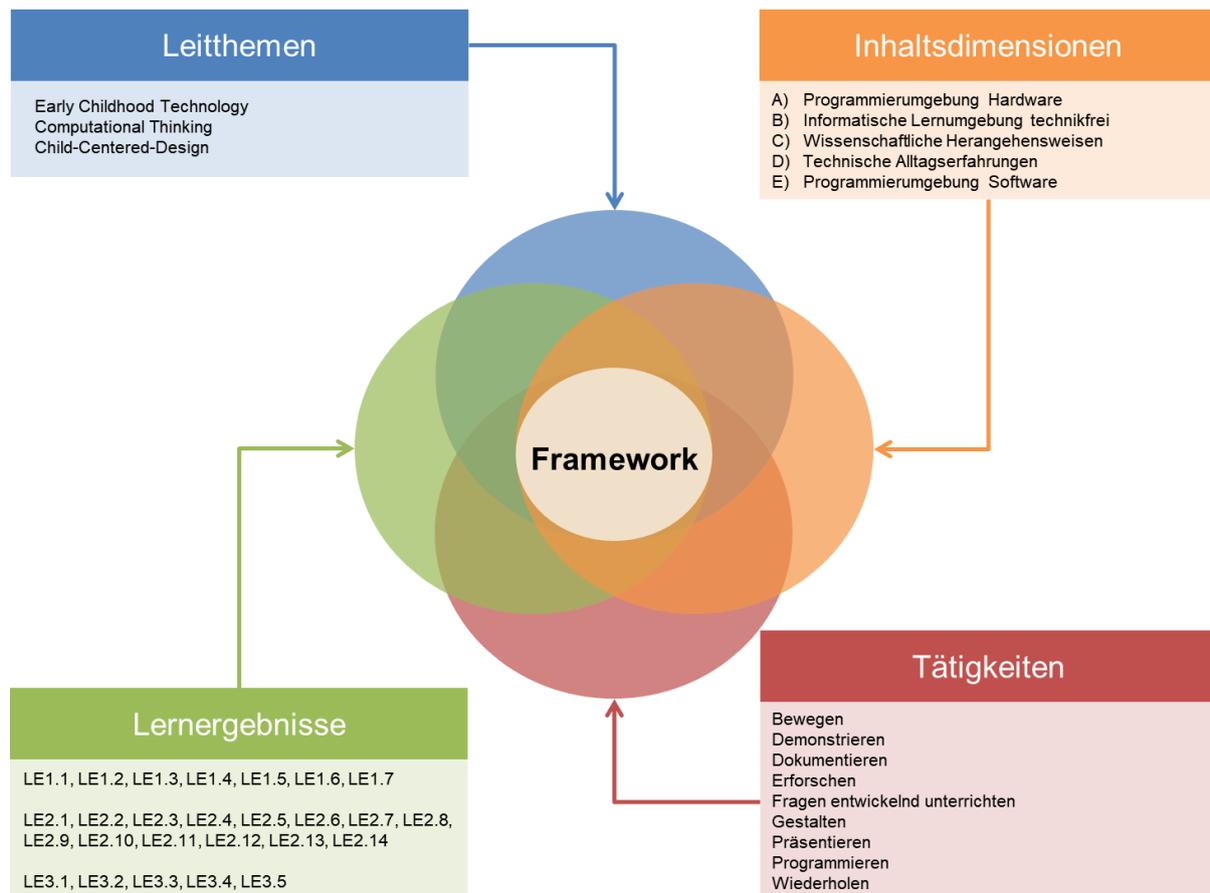


Abbildung 21: Framework - Überblick

8.7.1 Framework Perspektive 1: Aktivitätsschablonen nach Inhaltsdimensionen

Während der Entwicklung des Frameworks halfen die Inhaltsdimensionen bei der Kategorisierung und bei der Erstellung von thematischen Aufgabensammlungen. Im finalen Framework helfen die Inhaltsdimensionen bei der Ableitung von Aufgaben aus den Aktivitätsschablonen, da so die Möglichkeit besteht, inhaltlich selektiver und diverser vorzugehen, im Vergleich zu einer Ableitung von Aufgaben aus den Aktivitätsschablonen pro Leitthema. Es können somit auch in zeitlich kürzeren Lehr-Lern-Einheiten Inhalte vermittelt werden, die thematisch zusammenhängend und über die Leitthemen verteilt sind.

8.7.1.1 Inhaltsdimension A

AktivitätsschablonenID:	A1S1	Schwierigkeitsgrad:	1	Leitthema	Child-Centered-Design
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Erforschen			Dauer:	ca. 7 - 10 Minuten
Aktivitätsschablone:	<i>Die Kinder lernen eine Arbeitsmatte kennen, bevor ein Bodenroboter programmiert wird, der auf der Arbeitsmatte fahren soll.</i>				
Material:	Arbeitsmatte mit Symbolen				
Setting:	Bevor die Kinder den Bodenroboter zum Fahren auf der Arbeitsmatte programmieren, findet hier ein „Kennenlernen“ der Arbeitsmatte statt. Jedes Kind darf sich reihum ein Symbol aussuchen, das ihm am besten gefällt und erzählen was es darstellt. Daran anknüpfend kann als Überleitung zu weiteren Aktivitäten erwähnt werden, dass „der Bodenroboter jetzt auch die Matte kennenlernen möchte“.				
Ergänzende Informationen:	Falls kein Kind den generellen Aufbau der Arbeitsmatte in Form von gleich großen Kästchen erwähnt, diesbezüglich nachfragen. Daran kann angeknüpft werden, wenn der Bodenroboter auf der Arbeitsmatte programmiert wird.				

AktivitätsschablonenID:	A1S2	Schwierigkeitsgrad:	2	Leitthema	Child-Centered-Design
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>Die Kinder können durch freies Üben / Experimentieren die Funktionalität des eingesetzten Bodenroboters erforschen.</i>				
Material:	aufgeladene Bodenroboter, einer pro Kind oder pro zwei Kindern				
Setting:	Die Kinder bekommen Bodenroboter (möglicherweise in Schachteln zum Auspacken) ausgeteilt und werden zum Ausprobieren und Erkunden derselben aufgefordert.				
Ergänzende Informationen:	Nach einiger Zeit den Kindern, die Unterstützung beim Einschalten / ersten Bedienen des Bodenroboters benötigen, diese geben um Frustration vorzubeugen. Genügend großen Raum vorsehen. Kinder im „Tun“ versinken lassen.				

AktivitätsschablonenID:	A1S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 2 - 5 Minuten
Aktivitätsschablone:	<i>TR stellt Kind eine Aufgabe zur Programmierung eines Bodenroboters, um mit dem Bodenroboter von einem Kind zu einem anderen Kind (im Sitzkreis) zu fahren.</i>				
Material:	aufgeladenen Bodenroboter				
Setting:	Alle sitzen im Sitzkreis. Kinder bekommen reihum den Bodenroboter und bekommen je die Aufgabe mit dem Bodenroboter zu einem bestimmten anderen Kind im Sitzkreis zu fahren.				
Ergänzende Informationen:	Hier erwähnen / danach fragen, dass / ob die Schritte, die der Bodenroboter fährt immer gleich groß sind. Kinder planen am Boden die notwendigen Schritte bis zum Ziel. Manche Kinder behelfen sich hier mit im Raum vorhandenen Anhaltspunkten, wie z.B. gemasertem Parkettboden zum Abschätzen, wie viele Schritte notwendig sind. Wenn alle Kinder drankommen sollen, empfiehlt sich davor eine zwischenzeitliche Pause für WC-Besuch und Trinken. Außerdem benötigen die Kinder Unterstützung, wer nach der Reihe drankommen soll.				

AktivitätsschablonenID:	A1S4	Schwierigkeitsgrad:	4	Leitthema	Child-Centered-Design
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Präsentieren			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>Die Kinder erklären und zeigen (moderiert von TR) nicht mit dem Thema vertrauten Kindergartenpädagoginnen und -pädagogen die Funktionalität des eingesetzten Bodenroboters.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Alle sitzen im Sitzkreis, nicht mit dem Thema vertrauten Kindergartenpädagogin / befasster Kindergartenpädagogin sitzt mit im Kreis. Kinder erklären und zeigen gemeinsam (moderiert und unterstützt von TR) Kindergartenpäd. die Funktionalität des Bodenroboters. Darüber hinaus veranschaulichen sie durch exemplarisches Lösen von selbstgewählten Aufgaben die Einsatzmöglichkeiten des Bodenroboters.				
Ergänzende Informationen:	TR ermuntert die Kinder ihre Handlungen zu kommentieren (Talk Aloud).				

AktivitätsschablonenID:	A2S3	Schwierigkeitsgrad:	3	Leitthema	Child-Centered-Design
Aktivitätscluster:	A2) Vorstellung Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>TR stellt den zum Einsatz kommenden Bodenroboter vor, erklärt seine Funktionalität und demonstriert diese.</i>				

8 Empirische Entwicklung eines Frameworks

Material:	aufgeladener Bodenroboter, Arbeitsmatte
Setting:	Alle sitzen im Sitzkreis. TR zeigt und erklärt den eingesetzten Bodenroboter und führt ihn dann vor. TR arbeitet heraus und veranschaulicht, dass derselbe eingespeicherte Befehl bei Abarbeitung immer das selbe Verhalten des Bodenroboters bewirkt. Im Zuge dessen weist TR auch auf den Umstand hin, dass der Bodenroboter immer dieselbe Schrittweite und denselben Drehwinkel ausführt und zeigt dies beim Fahren des Bodenroboters auf der Arbeitsmatte.
Ergänzende Informationen:	Im Zuge der Erklärungen der Befehlstasten sollen auch Hilfestellungen für Richtungsunterscheidungen gegeben werden - z.B. Vorwärts ist dort, wo der Bodenroboter hinschaut, Unterscheidung links und rechts erarbeiten. Auf Nachfrage, ob die Kinder diese beiden Richtungen schon kennen antworten die meisten Kinder mit Ja. Sie wissen links / rechts von oben / unten zu unterscheiden, haben allerdings oftmals noch Probleme bei der Unterscheidung von links und rechts. Kinder auf Merkmale wie Armbänder, etc. als Anker hinweisen um ihnen die Unterscheidung zu erleichtern.

AktivitätsschablonenID:	A2S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	A2) Vorstellung Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren			Dauer:	ca. 20 Minuten
Aktivitätsschablone:	<i>TR stellt den zum Einsatz kommenden Bodenroboter vor, erklärt und demonstriert seine Funktionalität mit Hilfe von Befehlskarten, die die möglichen Befehle, die dem Bodenroboter programmiert werden können, repräsentieren.</i>				
Material:	aufgeladener Bodenroboter, Befehlskarten-Set mit den möglichen Befehlen des Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	Alle sitzen im Sitzkreis. Zeigen und vorführen des Bodenroboters durch TR. Anschließend erklären und demonstrieren der Funktionalität des Bodenroboters durch TR. Parallel zur Erklärung der unterschiedlichen Befehle stellt TR die zugehörige Befehlskarte vor, die den gespeicherten Befehl repräsentiert. In weiterer Folge visualisieren des Speicherinhalts des Bodenroboters durch Auflegen der den gespeicherten Befehlen entsprechenden Befehlskarten am Boden. Beim Auflegen der Befehlskarten empfiehlt sich ein Vorgehen von links nach rechts, wobei ganz links die Befehlskarte gelegt wird, die den als erstes eingespeicherten Befehl visualisiert und ganz rechts die Befehlskarte platziert wird, die den zuletzt gespeicherten Befehl anzeigt.				
Ergänzende Informationen:	Auf den Befehlskarten wird der jeweilige Befehl mit dem Symbol, durch welches er am Bodenroboter repräsentiert wird, dargestellt. Außerdem ist der Bodenroboter abgebildet um anzuzeigen in welche Richtung der Bodenroboter blickt und dass z.B. durch den Pfeil der einen Befehlskarte ein Vorwärtsbefehl und durch den Pfeil der anderen Befehlskarte ein Rückwärtsbefehl repräsentiert wird.				

AktivitätsschablonenID:	A3S1	Schwierigkeitsgrad:	1	Leitthema	Computational Thinking
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 2 Minuten
Aktivitätsschablone:	<i>TR stellt Kind eine Aufgabe zur Programmierung eines Bodenroboters ohne Richtungsänderung.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.				

AktivitätsschablonenID:	A3S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 2 - 3 Minuten
Aktivitätsschablone:	<i>TR stellt Kind eine Aufgabe zur Programmierung eines Bodenroboters mit einer Richtungsänderung.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.
----------------------------------	--

AktivitätsschablonenID:	A3S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 3 Minuten
Aktivitätsschablone:	<i>TR stellt Kind eine Aufgabe zur Programmierung eines Bodenroboters mit zwei Richtungsänderungen.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt. Ansprechen, dass es mehrere Wege von A nach B gibt und Kind dazu anregen, vor dem Einspeichern der Befehle den geplanten Weg anzugeben.				

AktivitätsschablonenID:	A3S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 3 - 5 Minuten
Aktivitätsschablone:	<i>TR stellt Kind eine Aufgabe zur Programmierung eines Bodenroboters mit mind. drei Richtungsänderungen.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt. Ansprechen, dass es mehrere Wege von A nach B gibt und Kind dazu anregen, vor dem Einspeichern der Befehle den geplanten Weg anzugeben.				

AktivitätsschablonenID:	A4S1	Schwierigkeitsgrad:	1	Leitthema	Computational Thinking
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 2 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung keine Richtungsänderung des Bodenroboters notwendig ist.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bodenroboter auf der Arbeitsmatte befahren möchte. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe keine Richtungsänderung programmiert werden muss. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.				

AktivitätsschablonenID:	A4S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 3 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung nur eine Richtungsänderung des Bodenroboters notwendig ist.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bodenroboter auf der Arbeitsmatte befahren möchte. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe nur eine Richtungsänderung programmiert werden muss. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.				

AktivitätsschablonenID:	A4S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 3 - 5 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung zwei Richtungsänderungen des Bodenroboters notwendig sind.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bodenroboter auf der Arbeitsmatte befahren möchte. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe zwei Richtungsänderungen programmiert werden müssen. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.				

AktivitätsschablonenID:	A4S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 3 - 5 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung mind. drei Richtungsänderungen des Bodenroboters notwendig sind.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				
Setting:	Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bodenroboter auf der Arbeitsmatte befahren möchte. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe mind. drei Richtungsänderungen programmiert werden müssen. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Ergänzende Informationen:	Arbeitsmatte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder, wenn Probleme mit dem Perspektivenwechsel zwischen Kind und Bodenroboter auftreten: sich direkt hinter den Bodenroboter auf die Arbeitsmatte (oder außen nahe zur Arbeitsmatte hinter den Bodenroboter) zu setzen, damit die Blickrichtung des einspeichernden Kindes und des Bodenroboters übereinstimmen und so die Wegplanung leichter fällt.				

AktivitätsschablonenID:	A5S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	A5) Gestelltes Problem lösen: Schleifen			Sozialform:	Ganze Gruppe
Aufgabenart:	Programmieren			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>TR stellt eine Aufgabe zur Programmierung eines Bodenroboters zum Fahren einer Figur mit einem sich wiederholenden Muster, auf einer Arbeitsmatte, um die Grundlagen des Schleifenkonzepts zu vermitteln.</i>				
Material:	aufgeladener Bodenroboter, Arbeitsmatte				

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Setting:	TR gibt den Weg vor, den der Bodenroboter auf der Arbeitsmatte befahren soll. Es handelt sich dabei um einen Weg mit einem sich wiederholenden Muster. TR gibt die Befehle ein, die die Kindergruppe gemeinsam erarbeiten. TR unterstützt als Moderatorin / Moderator und hilft mit Fragen, falls erforderlich. Im Anschluß bespricht TR mit den Kindern den Aufbau des programmierten Weges. Führt mit Fragen heran, dass sich Wegteile wiederholen und knüpft hier an um das Schleifenkonzept zu besprechen.
Ergänzende Informationen:	Ein möglicher Weg wäre mit dem Bodenroboter am Außenrand der Matte entlang zu fahren, also ein Quadrat oder ein Rechteck (je nach Form der Matte) zu programmieren oder den Bodenroboter jede „Spalte“ der Arbeitsmatte erkunden zu lassen.

AktivitätsschablonenID:	A6S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	A6) Selbstgewähltes Problem lösen: Schleifen			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>TR erarbeitet mit den Kindern die Grundlagen des Schleifenkonzepts anknüpfend an ihre Erfahrungen mit Bodenrobotern.</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Alle sitzen im Sitzkreis. Besprechen der Erkenntnisse der Kinder beim Lösen unterschiedlicher Aufgaben mit Bodenrobotern. Heranführen mit Nachfragen bezüglich der Aneinanderreihung der programmierten Befehle an die Begriffe "Wiederholungen", "Schleifenkonzept" bzw. auch "Algorithmusbegriff". Hilfreiche Fragen in diesem Zusammenhang könnten sein: "Wie kann ich den Weg rund um ein Quadrat zerlegen?", "Welche Teile des Weges sind gleich?", "Wie schaut der Weg aus, wenn ich ein zweites / drittes / viertes Mal rund um das Quadrat fahren möchte?"				
Ergänzende Informationen:	Mit dieser Aktivitätsschablone können den Kindern die Grenzen der Programmiermöglichkeiten eines einfachen Bodenroboters, wie z.B. des Bee-Bots, aufgezeigt werden. Mögliche Leitfragen zur Erörterung der Themen Wiederholung (Schleifen) können sein: Der Bodenroboter soll 20 Schritte vorwärts gehen – welche Möglichkeiten zur Programmierung gibt es dafür beim Bodenroboter? Wie könnte das anders gelöst werden – unabhängig vom Bodenroboter? Wie kann man mit möglichst wenigen Befehlsspeicherungen den Bodenroboter 5 Mal im Kreis „tanzen“ lassen? Der Bodenroboter soll dreimal um ein Hindernis fahren – wie würdest du das mit möglichst wenigen Befehlen programmieren?				

AktivitätsschablonenID:	A6S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	A6) Selbstgewähltes Problem lösen: Schleifen			Sozialform:	Kleingruppe
Aufgabenart:	Programmieren			Dauer:	ca. 20 Minuten
Aktivitätsschablone:	<i>Kinder stellen sich selbst eine Aufgabe zur Programmierung eines Bodenroboters zum Fahren rund um ein Hindernis aus ihrem Kindergartenumfeld in Form einer geometrischen Figur (z.B. Quadrat oder Rechteck) oder entlang eines Weges mit einem sich wiederholenden Muster. TR achtet bei der Aufgabenfindung durch die Kindergruppe darauf, dass diese zur Vermittlung der Grundlagen des Schleifenkonzepts geeignet ist.</i>				
Material:	aufgeladene Bodenroboter in ausreichender Anzahl – mindestens einen für zwei Kinder, geometrische Hindernisse in ausreichender Anzahl (Sitzpolster, Sitzteppich, Sessel, Hocker, etc.) bzw. Material zum Erstellen von Hinderniswegen				
Setting:	Die Kinder suchen sich in der Kleingruppe ein geometrisches Hindernis aus ihrem Kindergartenumfeld aus oder ordnen Hindernisse derart an, dass sich ein Weg mit einem sich wiederholenden Muster ergibt. Aufgabe ist es, in der Kleingruppe den Bodenroboter so zu programmieren, dass dieser erst einmal, dann zweimal, dreimal rund um das gewählte Hindernis oder rund um die Hindernisse des erbauten Hindernisweges mit sich wiederholendem Muster fährt. TR geht zwischen den Kleingruppen, die im Raum verteilt an dieser Aufgabe arbeiten, herum und unterstützt mit Hinweisen und falls erforderlich mit Hilfestellungen. Dabei führt TR in weiterer Folge in den Kleingruppen mit Fragen zum Schleifenkonzept.				
Ergänzende Informationen:	Sollten bei den Kindern in der Kleingruppe Probleme beim Programmieren des Bodenroboters rund um ihr gewähltes Hindernis auftreten, quadratisches Hindernis anbieten. Rechteck als Steigerung der Schwierigkeit, regelmäßiger Slalom oder ähnlicher Weg mit sich wiederholendem Muster weitere Steigerung der Schwierigkeit. Bei schnelleren Kleingruppen weitere Hindernisse mit steigender Schwierigkeit zur neuerlichen Programmierung vorstellen.				

8.7.1.2 Inhaltsdimension B

AktivitätsschablonenID:	B1S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	B1) Verbalisierte elementare Verarbeitungsschritte			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 15 - 20 Minuten
Aktivitätsschablone:	<i>Ein Kind spielt Roboter und wird durch verbale Befehlseingaben durch ein zweites Kind „gesteuert“.</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Die Kinder sollen sich in 2er-Teams zusammenfinden und führen die Aufgabe in den 2er-Teams				

	nacheinander in der Großgruppe durch. Ein Kind des 2er-Teams spielt Roboter und wird durch verbale Befehlseingaben vom anderen Kind gesteuert. Das Kind, das den Roboter spielt, führt nur genau das, was angesagt wird, ohne Interpretation und Ergänzungen aus. Die gerade nicht ins Roboter-Spiel involvierten Kinder sitzen im Sitz-Halbkreis und unterstützen mit Hilfestellungen, falls erforderlich.
Ergänzende Informationen:	Sollte von TR eindeutig exemplarisch vorgezeigt werden um u.a. die Vorgehensweise und notwendige Genauigkeit beim Formulieren und Ausführen der Befehle zu verdeutlichen. Bei der Erklärung der Aufgabe darauf eingehen, dass ein Roboter programmiert wird und nur genau die Befehle, die eingespeichert werden, exakt ausführt ohne Ergänzungen und Interpretationen vorzunehmen. Unterschied zum Menschen, dessen Gehirn die einzelnen Körperfunktionen steuert, erklären – Bogen spannen zu dieser Aufgabe, wo im 2er-Team das Kind, das die Anweisungen gibt, das andere Kind, das „Roboter“ spielt, bezüglich der Bewegungen programmiert.

AktivitätsschablonenID:	B2S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	B2) Gestelltes Problem lösen: visualisierte Sequenzen			Sozialform:	ganze Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 3 - 5 Minuten
Aktivitätsschablone:	<i>TR legt zur Demonstration der Zerlegung eines Algorithmus in Verarbeitungsschritte Befehlskarten am Boden auf und die Kinder bewegen sich nach den Vorgaben wie Roboter.</i>				
Material:	Befehlskarten-Set mit den möglichen Befehlen eines für die Zielgruppe geeigneten Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	TR legt am Boden eine einfache Abfolge von Befehlskarten von links (erster Befehl) nach rechts (letzter Befehl) auf, und erklärt den Kindern dass sie selbst diese Befehle als Roboter abarbeiten sollen. Dazu sollen die Kinder aufstehen, sich in einer Reihe mit genügend Abstand nebeneinander aufstellen und gesteuert durch ein akustisches Signal (z.B. klatschen) einen Befehl pro Signal abarbeiten, wobei TR mit gleicher Blickrichtung vor den Kindern steht und die Bewegung vorzeigt.				
Ergänzende Informationen:	Beschränkung auf Drehbewegungen bis Kinder sicher in der Umsetzung der auf der Befehlskarte abgebildeten Befehle in eigene Bewegungen sind beugt Chaos vor. Variation durch Inkludieren von Befehlskarten mit abgebildeten Vorwärts- und Rückwärtsbefehlen nach entsprechendem Übungsgrad. Auf genügend Platz rund um jedes Kind achten.				

AktivitätsschablonenID:	B2S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	B2) Gestelltes Problem lösen: visualisierte Sequenzen			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>Kinder legen eine beliebige Befehlsabfolge durch Einsatz von Befehlskarten am Boden auf und TR demonstriert durch "Theater-Spiel" die Abarbeitung der Befehle „im Speicher von TR“ (= Menge der aufgelegten Befehlskarten) und deren Auswirkungen.</i>				
Material:	Befehlskarten-Set mit den möglichen Befehlen eines für die Zielgruppe geeigneten Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	Kinder sitzen im Halbkreis. TR spielt einen Roboter, der durch aufgelegte Befehlskarten gesteuert wird. Die Kinder legen die Befehlskarten mit den abgebildeten Befehlen eines für die Zielgruppe geeigneten Bodenroboters am Boden in einer Reihe von links nach rechts auf. Links liegt der zuerst eingespeicherte Befehl, rechts der zuletzt eingespeicherte, der auch als letztes abgearbeitet wird. TR arbeitet die aufgelegten Befehle als Roboter ab und ein Kind der Gruppe zeigt als Veranschaulichung zusätzlich immer auf die Befehlskarte, die den aktuell abgearbeiteten Befehl darstellt.				
Ergänzende Informationen:	Auch das Auflegen von Befehlskarten zulassen, die keine sichtbare Reaktion von TR in der Rolle des Roboters bewirken (z.B. die Befehlskarte mit dem Befehl zum Start der Abarbeitung aller Befehle als erste aufgelegte Karte, oder Platzierung der Befehlskarte mit dem Befehl für Löschen unmittelbar vor der letzten Befehlskarte mit dem Befehl für den Start der Abarbeitung der Befehle). TR bespricht mit den Kindern im Anschluß an die Abarbeitung die Auswirkung der Aneinanderreihung der Befehlskarten.				

AktivitätsschablonenID:	B3S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 3 Minuten
Aktivitätsschablone:	<i>Ein Kind spielt Roboter und simuliert die Abarbeitung einer Befehlsabfolge, die von einem anderen Kind durch Auflegen von Befehlskarten vorgegeben wird. TR achtet beim Auflegen der Befehlskarten durch das Kind darauf, dass zur Lösung nur eine Richtungsänderung des Roboter spielenden Kindes notwendig ist.</i>				
Material:	Befehlskarten-Set mit den möglichen Befehlen eines für die Zielgruppe geeigneten				

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Setting:	Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen
	Die Kinder führen die Aktivität in 2er-Teams vor der ganzen Gruppe aus, die anderen Kinder sitzen im Halbkreis. Ein Kind des 2er-Teams überlegt sich den Weg, den das andere Kind als „Roboter“ im Raum zurücklegen soll und legt die korrespondierenden Befehlskarten von links (zuerst zu speichernder Befehl) nach rechts (zuletzt zu speichernder Befehl) auf dem Boden auf, die dem „Kind-Roboter“ für diesen Weg gespeichert werden müssen. Die anderen Kinder und TR unterstützen, falls erforderlich.
Ergänzende Informationen:	Ausreichend freie Fläche für das Auflegen und die Abarbeitung der Befehle sicherstellen. Das Roboter spielende Kind darauf hinweisen immer gleich große Schritte und Drehungen auszuführen und die in Form der Befehlskarten aufgelegten Befehle langsam einzeln abzuarbeiten.

AktivitätsschablonenID:	B3S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 3 Minuten
Aktivitätsschablone:	<i>Ein Kind spielt Roboter und simuliert die Abarbeitung einer Befehlsabfolge, die von einem anderen Kind durch Auflegen von Befehlskarten vorgegeben wird. TR achtet beim Auflegen der Befehlskarten durch das Kind darauf, dass die Befehlsabfolge zwei Richtungsänderungen beinhaltet.</i>				
Material:	Befehlskarten-Set mit den möglichen Befehlen eines für die Zielgruppe geeigneten Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	Die Kinder führen die Aktivität in 2er-Teams vor der ganzen Gruppe aus, die anderen Kinder sitzen im Halbkreis. Ein Kind des 2er-Teams überlegt sich den Weg, den das andere Kind als „Roboter“ im Raum zurücklegen soll und legt die korrespondierenden Befehlskarten von links (zuerst zu speichernder Befehl) nach rechts (zuletzt zu speichernder Befehl) auf dem Boden auf, die dem „Kind-Roboter“ für diesen Weg gespeichert werden müssen. Die anderen Kinder und TR unterstützen, falls erforderlich.				
Ergänzende Informationen:	Ausreichend freie Fläche für das Auflegen und die Abarbeitung der Befehle sicherstellen. Das Roboter spielende Kind darauf hinweisen immer gleich große Schritte und Drehungen auszuführen und die in Form der Befehlskarten aufgelegten Befehle langsam einzeln abzuarbeiten				

AktivitätsschablonenID:	B3S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 3 Minuten
Aktivitätsschablone:	<i>Ein Kind spielt Roboter und simuliert die Abarbeitung einer Befehlsabfolge, die von einem anderen Kind durch Auflegen von Befehlskarten vorgegeben wird. TR achtet beim Auflegen der Befehlskarten durch das Kind darauf, dass die Befehlsabfolge mind. drei Richtungsänderungen beinhaltet.</i>				
Material:	Befehlskarten-Set mit den möglichen Befehlen eines für die Zielgruppe geeigneten Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	Die Kinder führen die Aktivität in 2er-Teams vor der ganzen Gruppe aus, die anderen Kinder sitzen im Halbkreis. Ein Kind des 2er-Teams überlegt sich den Weg, den das andere Kind als „Roboter“ im Raum zurücklegen soll und legt die korrespondierenden Befehlskarten von links (zuerst zu speichernder Befehl) nach rechts (zuletzt zu speichernder Befehl) auf dem Boden auf, die dem „Kind-Roboter“ für diesen Weg gespeichert werden müssen. Die anderen Kinder und TR unterstützen, falls erforderlich.				
Ergänzende Informationen:	Ausreichend freie Fläche für das Auflegen und die Abarbeitung der Befehle sicherstellen. Das Roboter spielende Kind darauf hinweisen immer gleich große Schritte und Drehungen auszuführen und die in Form der Befehlskarten aufgelegten Befehle langsam einzeln abzuarbeiten				

AktivitätsschablonenID:	B4S1	Schwierigkeitsgrad:	1	Leitthema	Computational Thinking
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	Einzel
Aufgabenart:	Bewegen			Dauer:	ca. 5 - 7 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen, ohne Einsatz von Befehlskarten und Vorgaben, Robotertheater spielen.</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Die Kinder sollen sich frei im Raum bewegen und spielen, dass sie ein Roboter sind (Bewegungen, Laute, etc.).				
Ergänzende Informationen:	Auf Raumgröße und freie Fläche achten. Die Kinder vor dieser Übung an gewaltfreies Spiel erinnern, Lärmentwicklung durch kurze Dauer begrenzen.				

AktivitätsschablonenID:	B4S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen			Dauer:	ca. 7 Minuten
Aktivitätsschablone:	<i>Wiederholung der möglichen Befehle zur Programmierung des eingesetzten Bodenroboters durch TR anhand von Befehlskarten.</i>				
Material:	ein Befehlskarten-Set mit den möglichen Befehlen des eingesetzten Bodenroboters jeweils auf einer (Karton-)Karte (z.B. im Format A6), wobei alle Befehle mehrfach vorhanden sein müssen				
Setting:	Kinder sitzen im Halbkreis. TR hat die verschiedenen Befehlskarten gut sichtbar auf dem Boden vor den Kindern aufgelegt. Die Befehlskarten zeigen den abgebildeten Befehl als Symbol, das den Befehl auch am Bodenroboter repräsentiert und sind zusätzlich mit dem jeweiligen Befehl beschriftet. Außerdem ist visuell dargestellt welche Kante des Kärtchens die obere ist (oder welche Ecke die linke obere ist) um die eindeutige Interpretation des dargestellten Befehls zu gewährleisten. Anknüpfend an die Wiedererkennung der Befehle auf den Befehlskarten durch die Kinder wiederholt TR die Befehle des eingesetzten Bodenroboters anhand der Befehlskarten.				
Ergänzende Informationen:	Keine				

AktivitätsschablonenID:	B4S3	Schwierigkeitsgrad:	3	Leitthema	Computational Thinking
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten			Dauer:	ca. 10 Minuten
Aktivitätsschablone:	<i>TR erläutert und bespricht mit den Kindern den Begriff „Programmieren“ und dessen grundlegende Konzepte, dazu gehören u.a.: -) Elementare Verarbeitungsschritte -) Sequenzen</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Alle sitzen im Sitzkreis. Ausgehend von den Vorerfahrungen aus dem Alltag der Kinder besprechen der Begriffe „programmieren“, „speichern“, „ausführen“, „löschen“ und „Befehl“ sowie „elementarer Verarbeitungsschritt“ und „Sequenz“.				
Ergänzende Informationen:	Kurz und prägnant halten und möglichst auf Alltagswissen der Kinder (z.B. Waschmaschine vorprogrammieren, Lift holen, Garagentor öffnen, Speicherkarte einer Digitalkamera sichern, eine App installieren, etc.) aufbauen.				

AktivitätsschablonenID:	B4S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Gestalten			Dauer:	ca. 20 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen einen Algorithmus aus ihrem Alltag in visuell darstellbare Verarbeitungsschritte zerlegen und diese auf selbst zu gestaltende Algorithmenkärtchen (ähnlich einer Bildgeschichte) zeichnen.</i>				
Material:	Forschungshefte, (Moderations-)Kärtchen in ausreichender Anzahl (ca. 6 Stück pro Kind), Farbstifte in ausreichender Anzahl und in unterschiedlichen Farben, Spitzer, Radiergummi, Klarsichtfolien zum Sammeln der Kärtchen pro Kind für das Einkleben ins Forschungsheft				
Setting:	Alle sitzen im Sitzkreis. Es soll ein Algorithmus in visuell darstellbare Verarbeitungsschritte zerlegt und diese Verarbeitungsschritte in Form einer Art „Bildgeschichte“ jeweils auf ein eigenes Kärtchen gezeichnet werden. Die Kärtchen werden pro Kind in einer Klarsichtfolie gesammelt und diese von TR ins jeweilige Forschungsheft der Kinder eingeklebt.				
Ergänzende Informationen:	Als Hintergrundgeschichte zum Verständnis für die Kinder kann erzählt werden, dass ein Außerirdischer auf die Erde kommt und die Kinder diesem Dinge aus ihrem täglichen Leben lernen sollen. Als Beispiele können genannt werden sich in der Früh anziehen, etwas essen, eine Stiege raufgehen, schlafen gehen, etc. Die Kinder sollen sich für ein Thema entscheiden, das sie dem Außerirdischen lernen möchten und dieses in möglichst elementare, visuell darstellbare Verarbeitungsschritte zerlegt auf die Kärtchen zeichnen.				

8.7.1.3 Inhaltsdimension C

AktivitätsschablonenID:	C1S1	Schwierigkeitsgrad:	1	Leitthema	Early Childhood Technology
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten			Dauer:	ca. 10 Minuten
Aktivitätsschablone:	<i>TR erläutert den Begriff „Forschen“ und teilt Forschungshefte an die Kinder aus.</i>				
Material:	pro teilnehmendem Kind ein unliniertes Heft mit 20 Blatt, A4 oder Quart				
Setting:	Alle sitzen im Sitzkreis. TR erklärt den Begriff "Forschen" kurz kindgerecht und bindet die Kinder durch Fragen bezüglich ihrer Forschungserfahrungen ein.				
Ergänzende Informationen:	Anknüpfen an Erfahrung der Kinder aus Naturereignissen, bei der Beobachtung von Tieren / Pflanzen, beim Entdecken von Funktionalitäten von Spielzeug, beim Beobachten von physikalischen Gesetzen, etc.				

AktivitätsschablonenID:	C1S2	Schwierigkeitsgrad:	2	Leitthema	Child-Centered-Design
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel
Aufgabenart:	Gestalten			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen eine oder mehrere Zeichnungen von Robotern anfertigen.</i>				
Material:	unlinierte Blätter A4, mehr als ein Blatt pro teilnehmendem Kind, Blei- und Buntstifte in ausreichender Anzahl und Farbauswahl, Spitzer und Radiergummis				
Setting:	Die Kinder sollen einen oder mehrere Roboter zeichnen, wobei sie sowohl ihrer Fantasie freien Lauf lassen, als auch auf die Darstellung ihnen schon bekannter Roboter zurückgreifen dürfen.				
Ergänzende Informationen:	Die extra ausgeteilten unlinierten A4-Blätter, die anschließend ins Forschungsheft geklebt werden, verhindern ein Durchdrücken der Zeichnungen im Forschungsheft und ermöglichen bei mehreren Zeichnungen, dass nur das Best Of ins Forschungsheft geklebt werden kann. Eine Sitzanordnung im Sitzkreis dergestalt, dass die Kinder im Sitzkreis abwechselnd mit Blick nach innen und nach außen sitzen fördert die Kreativität und eigene Ideen bei den Kindern.				

AktivitätsschablonenID:	C1S3	Schwierigkeitsgrad:	3	Leitthema	Child-Centered-Design
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Präsentieren			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen von ihnen selbst erstellte Artefakte präsentieren.</i>				
Material:	selbsterstellte Artefakte der Kinder				
Setting:	Die Kinder sitzen im Sitzkreis und präsentieren reihum ihre selbsterstellten Artefakte. Die anderen Kinder werden von TR angehalten nachzufragen, was das selbsterstellte Artefakt kann, welcher Art es genau ist, warum gewisse Details in der vorliegenden Art und Weise gestaltet wurden, etc.				
Ergänzende Informationen:	Ohne entsprechende Nachfragen erzählen die meisten Kinder relativ wenig über die Funktionalität etc. ihrer selbst erstellten Artefakte, sondern eher Geschichten. Falls von anderen Kindern keine Fragen kommen, hier unterstützen.				

AktivitätsschablonenID:	C2S1	Schwierigkeitsgrad:	1	Leitthema	Early Childhood Technology
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen mit dem Forschungsheft als Forscherin bzw. Forscher arbeiten und:</i> -) das Forschungsheft und ein Namensschild mit Namen versehen (mit Unterstützung durch TR), -) selbstgemachte Zeichnungen von Robotern und -) selbsterstellte Fotos von technischen Geräten mit Informatikbezug und -) Bilder von Robotern und des zum Einsatz kommenden Bodenroboters einkleben				
Material:	Forschungshefte, Namensschilder, Blei- und Buntstifte, Spitzer und Radiergummi, Roboterzeichnungen der Kinder, von Kindern selbsterstellte Fotos von technischen Geräten mit Informatikbezug, Bilder von Robotern und Bilder des zum Einsatz kommenden Bodenroboters in ausreichender Anzahl, Klebstoff, der im Kindergarten verwendet werden darf, in ausreichender Anzahl				
Setting:	Alle Kinder sitzen im Sitzkreis. Die Kinder beschriften soweit sie es schon können selbst mit Vornamen oder Symbol das Forschungsheft und ihr Namensschild, TR hilft dabei. Sie kleben mit Unterstützung die Zeichnungen, Fotos und Bilder ein.				
Ergänzende Informationen:	Kinder beschriften mit Symbolen (wie im Kindergarten für Garderobenhaken, etc. gebräuchlich) oder Namen. Die meisten Kinder benötigen Hilfe beim Schreiben des Namens und beim Einkleben, z.B. auch bei der Platzierung der Zeichnungen, Bilder und Fotos im Forschungsheft.				

AktivitätsschablonenID:	C2S3	Schwierigkeitsgrad:	3	Leitthema	Early Childhood Technology
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren			Dauer:	ca. 10 Minuten

Aktivitätsschablone:	<i>Die Kinder sollen den Einsatz eines technischen Geräts mit Informatikbezug im Alltag (z.B. Staubsaugerroboter) mittels altersgerechten digitalen Fotoapparaten dokumentieren.</i>
Material:	aufgeladenes technisches Gerät mit Informatikbezug (wie z.B. Staubsaugerroboter), aufgeladene altersgerechte digitale Fotoapparate in ausreichender Anzahl
Setting:	Zuerst wird das technische Gerät mit Informatikbezug in Betrieb genommen und dann die altersgerechten digitalen Fotoapparate ausgeteilt. Die Aufgabe der Kinder ist es, den Einsatz des technischen Geräts mit Informatikbezug mit den Fotoapparaten zu kommentieren. TR unterstützt mit Hilfestellungen bezüglich der Vorgehensweise bei der Erstellung der Fotos, falls erforderlich.
Ergänzende Informationen:	Eher kurze Zeitspanne für die Aufgabe wählen um das Auffinden der Zusatzfunktionalitäten der altersgerechten digitalen Fotoapparate (wie etwa die Spielefunktion) zu umgehen.

AktivitätsschablonenID:	C2S4	Schwierigkeitsgrad:	4	Leitthema	Early Childhood Technology
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	ganze Gruppe
Aufgabenart:	Dokumentieren			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>Die Kinder besprechen mit TR anhand der Einträge im Forschungsheft die Chronologie der Ereignisse und lernen die Notwendigkeit einer Beschriftung und Datumsnennung kennen.</i>				
Material:	Forschungshefte, Bunt- und Filzstifte für TR und US um Datumseinträge in den Forschungsheften der Kinder vornehmen zu können				
Setting:	Alle sitzen im Sitzkreis. Besprechen der Chronologie der Ereignisse anhand der Einträge im Forschungsheft und der Beschriftungsnotwendigkeit von Forschungsergebnissen. Gemeinsames erarbeiten der Begriffe Jahr, Monat, Tag und besprechen, dass die Kombination aus diesen dreien das eindeutige Datum z.B. des heutigen Tages angibt. Beschriften der Einträge im Forschungsheft mit dem jeweiligen Datum des Ereignisses, das durch den Forschungsheft-Eintrag visualisiert wird.				
Ergänzende Informationen:	Das Datum selbst zu schreiben ist für die Kinder noch zu schwierig - daher die Beschriftung durch TR und US vornehmen und in den von den Kindern gewünschten Farben die jeweiligen Einträge in deren Forschungsheft mit dem entsprechenden Datum versehen.				

AktivitätsschablonenID:	C3S2	Schwierigkeitsgrad:	2	Leitthema	Child-Centered-Design
Aktivitätscluster:	C3) Konzeption und Erstellung Lernmaterial			Sozialform:	Kleingruppe
Aufgabenart:	Gestalten			Dauer:	ca. 25 - 30 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen Arbeitsmatten zur Nutzung mit dem gewählten Bodenroboter frei gestalten.</i>				
Material:	Blätter mit einem vorgezeichneten Raster in Schrittweite des Bodenroboters – je Kleingruppe von drei Kindern mindestens ein Blatt (z.B. 5 mal 4 Kästchen auf Flipchart- oder Backpapier), Farbstiftsets, Spitzer und Radiergummi für jede Kleingruppe, wiederablösbares Klebeband um die Mattenrohlinge am Boden festzukleben				
Setting:	Die Kinder sollen die Mattenrohlinge in Kleingruppen mit eigenen Ideen gestalten. Welche Welt soll der Bodenroboter auf der Arbeitsmatte erkunden können?				
Ergänzende Informationen:	Ausreichender Platz rund um die jeweilige Zeichnung (am Boden - die Kinder malen gern am Boden auf Polstern rund um die Matte sitzend) wird benötigt, um zu verhindern, dass die Kinder beim Holen von Stiften, Wechseln der Mattenseite, etc. auf die Matte steigen oder gar die Ecken umknicken / einreißen etc.				

AktivitätsschablonenID:	C3S4	Schwierigkeitsgrad:	4	Leitthema	Child-Centered-Design
Aktivitätscluster:	C3) Konzeption und Erstellung Lernmaterial			Sozialform:	Einzel
Aufgabenart:	Gestalten			Dauer:	ca. 30 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen einen Prototyp eines Bodenroboters basteln (durch Zeichnen, Ausschneiden, Kleben), mit dem andere Kinder Programmieren lernen können.</i>				
Material:	Bastelmaterialien in ausreichender Menge: A4 Blatt Papier für jedes Kind, zusätzliches Papier, Moosgummi in verschiedenen Farben, Stifte, Schere, Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Menge, Forschungsheft, Klarsichthüllen zur späteren Aufbewahrung / Sammlung				
Setting:	Auf dem ausgeteilten losen Blatt sollen sich die Kinder zuerst überlegen wie der Roboter aussehen kann, dann diese Vorlage ausschneiden, auf den Moosgummi legen, abpausen und aus dem Moosgummi nach der Vorlage ausschneiden. Bedienelemente aus Moosgummi extra ausschneiden und dann an der gewünschten Position auf die Moosgummi-Grundform kleben.				
Ergänzende Informationen:	Hilfestellung: zuerst Grundform des zu bastelnden Bodenroboters überlegen, aus der Vogelperspektive anschauen und aus dieser Perspektive zeichnen - veranschaulichen anhand von unterschiedlichen Objekten aus ihrem Kindergartenalltag. Nach Erklärung der Aufgabe durch TR empfiehlt sich eine Arbeitsdauer für die Kinder von ca. 20 Minuten, danach sinkt die Aufmerksamkeit deutlich. Sollte noch mehr Zeit benötigt werden, empfiehlt sich eine Weiterarbeit zu einem anderen Zeitpunkt. Hinweis auf die geeignete Größe des Prototyps, damit ein Kind, das mit einem auf diesem Prototyp basierenden Roboter Programmieren lernen würde, diesen gut bedienen könnte (Tastengröße, etc.). Kinder benötigen hohes Maß an Unterstützung mit Fragen: „Wie soll der Roboter bedient werden?“, „Wieviele Bedienelemente soll der Roboter bekommen?“, „Welche Funktionalitäten soll der Roboter haben?“, „Welche Farben eignen sich				

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

	für die unterschiedlichen Bedienelemente?“, etc. Diese Aufgabe ist für eine altersgemischte Kindergartengruppe (3 bis 6 Jahre) zu schwer in fast allen ihren Teilen - > für 6-jährige Kindergartenkinder (Vorschulkinder) bei geeigneter Vorarbeit geeignet, da hier das Unterstützen beim Ausschneiden, Kleben, Zeichnen an sich wegfällt und sich die Unterstützung auf Prototyp-bezogene Fragestellungen beschränkt.
--	---

AktivitätsschablonenID:	C4S2	Schwierigkeitsgrad:	2	Leitthema	Early Childhood Technology
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen			Dauer:	ca. 5 – 7 Minuten
Aktivitätsschablone:	<i>Die Kinder wiederholen die Funktionalität des eingesetzten Bodenroboters.</i>				
Material:	aufgeladener Bodenroboter				
Setting:	Alle sitzen im Sitzkreis. Gemeinsam mit den Kindern wiederholt TR die Funktionsweise des Bodenroboters. Vom Einschalten des Bodenroboters, über das Einspeichern von Befehlen, das Abarbeiten von Befehlen bis zum Löschen derselben, immer auch mit Veranschaulichung am Bodenroboter in Betrieb.				
Ergänzende Informationen:	Keine				

AktivitätsschablonenID:	C4S3	Schwierigkeitsgrad:	3	Leitthema	Early Childhood Technology
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>Kinder berichten in Form einer Reflexionsrunde von ihren Erlebnissen mit den eingesetzten Robotern.</i>				
Material:	aufgeladene eingesetzte Roboter				
Setting:	Alle sitzen im Sitzkreis. Kinder berichten reihum von ihren bisherigen Erlebnissen mit den eingesetzten Robotern. TR unterstützt den Reflexionsprozess durch Nachfragen.				
Ergänzende Informationen:	Das jeweilige Kind, das gerade von seinen Erlebnissen mit den eingesetzten Robotern berichtet, könnte sich aussuchen, welchen der Roboter es vor sich stehen haben möchte, in Händen halten möchte falls möglich, um das Abrufen der Erlebnisse zu unterstützen.				

8.7.1.4 Inhaltsdimension D

AktivitätsschablonenID:	D1S1	Schwierigkeitsgrad:	1	Leitthema	Early Childhood Technology
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren			Dauer:	ca. 10 Minuten
Aktivitätsschablone:	<i>Vorzeigen und erklären von Robotern aus der die Kinder umgebenden Umwelt anhand von Bildern durch TR.</i>				
Material:	Bilder von unterschiedlichen Robotern, geeignet sind Bilder von Haushaltsrobotern, Industrierobotern, Marsrobotern, Rettungsrobotern, Roboterautos, Robotern zur Bombenentschärfung und auch Robotern, die in Kind adäquaten Filmen vorkommen, z.B. Wolley, Forschungshefte, Bilder der Roboter in ausreichender Anzahl pro Motiv				
Setting:	TR zeigt und erklärt die Roboter auf den Bildern, von denen sich die Kinder je zwei Bilder aussuchen dürfen.				
Ergänzende Informationen:	Die Kinder sollen die ausgewählten Bilder später gesammelt mit anderem Material ins Forschungsheft einkleben, bis dahin verwahrt TR im jeweiligen Forschungsheft die ausgewählten Bilder.				

AktivitätsschablonenID:	D1S2	Schwierigkeitsgrad:	2	Leitthema	Early Childhood Technology
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>TR stellt unterschiedliche technische Geräte mit Bezug zur Informatik vor.</i>				
Material:	z.B. aufgeladener Staubsaugerroboter, mobiler Fotodrucker, Tablet, Mobiltelefon, etc.				
Setting:	TR erklärt die Funktionsweise des technischen Geräts mit Bezug zur Informatik zuerst kurz im ausgeschalteten Zustand. Dann wird das technische Gerät in Betrieb genommen, währenddessen geht TR auf hard- und softwaretechnische Aspekte des jeweiligen Geräts ein. Die Kinder dürfen das technische Gerät angreifen und bedienen.				
Ergänzende Informationen:	Durch Fragen an die Vorerfahrungen der Kinder anknüpfen.				

AktivitätsschablonenID:	D2S3	Schwierigkeitsgrad:	3	Leitthema	Early Childhood Technology
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten			Dauer:	ca. 15 Minuten
Aktivitätsschablone:	<i>TR befragt Kinder, wie sie sich Roboter vorstellen, ob sie schon einmal mit Robotern in Berührung gekommen sind und ob sie welche besitzen.</i>				
Material:	Keine besonderen Materialien erforderlich.				
Setting:	Alle sitzen im Sitzkreis. TR stellt Fragen und regt die Kinder zum Mitarbeiten an. Folgende Fragen können eine Orientierung bieten z.B.: „Habt ihr schon einmal einen Roboter gesehen?“, „Habt ihr schon einmal etwas von Robotern gehört?“, „Lebt ein Roboter?“ Zusätzliche Fragen zu den Themen: Wahrnehmung, „Nahrung“, „Sinnesorgane“, Fortbewegung und Einsatzzwecke von Robotern.				
Ergänzende Informationen:	Keine				

AktivitätsschablonenID:	D2S4	Schwierigkeitsgrad:	4	Leitthema	Child-Centered-Design
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten			Dauer:	ca. 10 - 15 Minuten
Aktivitätsschablone:	<i>TR befragt Kinder, ob sie schon einmal mit einem Computer in Berührung gekommen sind und wie sie denken, dass Computer funktionieren.</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Alle sitzen im Sitzkreis. TR moderiert ein Gruppengespräch über Computer. TR fragt nach den Erlebnissen der Kinder mit Computern. Es werden Unterschiede zu Menschen und Gegenständen des täglichen Gebrauchs herausgearbeitet. In groben Zügen wird die Funktionsweise von Computern besprochen.				
Ergänzende Informationen:	Diese Aufgabe eignet sich unter anderem gut dazu, die Vorerfahrungen der Kinder mit Computern in Erfahrung zu bringen.				

AktivitätsschablonenID:	D3S2	Schwierigkeitsgrad:	2	Leitthema	Early Childhood Technology
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte			Sozialform:	Einzel
Aufgabenart:	Erforschen			Dauer:	ca. 4 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen sich durch Ausprobieren mit der Funktionalität von altersgerechten digitalen Fotoapparaten vertraut machen.</i>				
Material:	aufgeladene altersgerechte digitale Fotoapparate für jedes Kind				
Setting:	Kinder dürfen die altersgerechten digitalen Fotoapparate frei erkunden und ausprobieren.				
Ergänzende Informationen:	Eher kurze Zeitspanne für die Aufgabe wählen um das Auffinden der Zusatzfunktionalitäten wie Spiele zu umgehen. Die Fotoapparate könnten auch in 2er-Teams erkundet werden, wenn weniger Geräte zur Verfügung stehen, dann ein paar Minuten länger Zeit geben.				

AktivitätsschablonenID:	D3S3	Schwierigkeitsgrad:	3	Leitthema	Early Childhood Technology
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Dokumentieren			Dauer:	ca. 15 - 20 Minuten
Aktivitätsschablone:	<i>Die Kinder drucken mit Unterstützung durch TR die Fotos, die mit altersgerechten digitalen Fotoapparaten erstellt wurden, aus.</i>				
Material:	Fotodrucker, der ohne Verbindung zum Computer direkt von Speicherkarte oder durch Verbindung zur Kamera mit USB-Kabel Fotos drucken kann, ausreichend Fotopapier für diesen Drucker, vergewissern dass Patronen genügend gefüllt sind				
Setting:	Alle sitzen im Sitzkreis. Jedes Kind darf reihum ein Foto ausdrucken, das mit den altersgerechten digitalen Fotoapparaten vom Einsatz eines technischen Geräts mit Informatikbezug im Alltag gemacht wurde.				
Ergänzende Informationen:	Das Funktionsprinzip des verwendeten Druckers kann hier während des Ausdrucks erklärt werden (z.B. wenn das Foto immer wieder eingezogen wird und sich die Farben des Fotos langsam aufbauen).				

8.7.1.5 Inhaltsdimension E

AktivitätsschablonenID:	E1S4	Schwierigkeitsgrad:	4	Leitthema	Child-Centered-Design
Aktivitätscluster:	E1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen			Dauer:	ca. 7 - 10 Minuten
Aktivitätsschablone:	<i>Kind soll die Funktionsweise der Steuerung einer Simulation eines Bodenroboters erfassen und die Simulation sinngemäß nutzen.</i>				

8.7 Phase 6: Finalisierung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“

Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind		
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der Simulation eines Bodenroboters und wird z.B. gefragt: „Wozu denkst du dienen die verschiedenen Bedienelemente des Programms?“, „Wie kann der Bodenroboter in der Simulation gesteuert werden?“, „Welche Aufgaben soll der Bodenroboter im Programm lösen?“, „Welche Aufgabe möchtest du versuchen zu lösen?“		
Ergänzende Informationen:	TR regt Kind durch Nachfragen an, sich selbst über die Bedienung Gedanken zu machen und an schon Bekanntes aus der Bedienung von Computer-Spielen oder auch aus der Steuerung realer Bodenroboter oder anderer Lernsoftware anzuknüpfen.		

AktivitätsschablonenID:	E2S1	Schwierigkeitsgrad:	1	Leitthema	Early Childhood Technology
Aktivitätscluster:	E2) Vorstellung Programmierumgebung		Sozialform:	Einzel	
Aufgabenart:	Demonstrieren		Dauer:	ca. 7 Minuten	
Aktivitätsschablone:	<i>TR stellt dem Kind die Simulation eines Bodenroboters vor und demonstriert ihm die Bedienung.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der Simulation eines Bodenroboters. TR stellt dem Kind den Aufbau der Simulation vor und veranschaulicht die Funktionsweise anhand der Programmierung eines ersten einfachen Programms.				
Ergänzende Informationen:	Keine				

AktivitätsschablonenID:	E3S3	Schwierigkeitsgrad:	3	Leitthema	Child-Centered-Design
Aktivitätscluster:	E3) Wissenstransfer		Sozialform:	Einzel	
Aufgabenart:	Erforschen		Dauer:	ca. 5 - 7 Minuten	
Aktivitätsschablone:	<i>Kind soll sich in der Simulation eines Bodenroboters orientieren und dabei an themenrelevante Vorerfahrungen anknüpfen.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der Simulation eines Bodenroboters und wird gebeten: „Bitte kannst du mir beschreiben was du siehst“. Damit soll erreicht werden, dass sich das Kind mit den verschiedenen Elementen der Oberfläche der Software beschäftigt und eventuell schon überlegt wie die Software funktioniert. Es könnte noch nachgefragt werden: „Kommt dir irgendetwas im Programm bekannt vor?“ bzw. „Wozu dient deiner Meinung nach dieses Programm?“				
Ergänzende Informationen:	Keine				

AktivitätsschablonenID:	E3S4	Schwierigkeitsgrad:	4	Leitthema	Early Childhood Technology
Aktivitätscluster:	E3) Wissenstransfer		Sozialform:	Einzel	
Aufgabenart:	Erforschen		Dauer:	ca. 7 - 10 Minuten	
Aktivitätsschablone:	<i>Kind soll Unterschiede zwischen einer Simulation eines Bodenroboters und einem eingesetzten realen Bodenroboter benennen und bestimmte definierte Funktionstasten in der Simulation auffinden können.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der Simulation eines Bodenroboters und wird z.B. gefragt: „Findest du Unterschiede zwischen dem von uns programmierten Bodenroboter und dem Bodenroboter hier im Programm, vielleicht bei den Buttons?“, „Denkst du, dass der Bodenroboter hier im Programm ähnliche Dinge kann, wie der von uns programmierte Bodenroboter?“, „Was gefällt dir besser?“, „Wo denkst du befindet sich in der Simulation der Button für Start?“				
Ergänzende Informationen:	Dem Kind die Steuerung des Programms überlassen, sodass das Kind auch durch Ausprobieren Unterschiede, falls vorhanden, und bestimmte definierte Funktionstasten auffinden kann.				

AktivitätsschablonenID:	E4S1	Schwierigkeitsgrad:	1	Leitthema	Computational Thinking
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen		Sozialform:	Einzel	
Aufgabenart:	Programmieren		Dauer:	ca. 2 Minuten	
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung einer Simulation eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung keine Richtungsänderung der Simulation des Bodenroboters notwendig ist.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch				

Setting:	und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm einer Simulation eines Bodenroboters und darf sich selbst eine Aufgabe zur Programmierung der Simulation des Bodenroboters stellen. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe keine Richtungsänderung programmiert werden muss. Das Kind programmiert den Weg ein, TR unterstützt es dabei, falls erforderlich.
Ergänzende Informationen:	Die Steuerung des Bodenroboters in der Simulation sollte entweder schon besprochen worden sein oder eine Besprechung im Zuge dieser Aktivität erfolgen, dann Dauer anpassen.

AktivitätsschabloneID:	E4S2	Schwierigkeitsgrad:	2	Leitthema	Computational Thinking
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren			Dauer:	ca. 2 - 3 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung einer Simulation eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung nur eine Richtungsänderung der Simulation des Bodenroboters notwendig ist.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm einer Simulation eines Bodenroboters und darf sich selbst eine Aufgabe zur Programmierung der Simulation des Bodenroboters stellen. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe nur eine Richtungsänderung programmiert werden muss. Das Kind programmiert den Weg ein, TR unterstützt es dabei, falls erforderlich.				
Ergänzende Informationen:	Die Steuerung des Bodenroboters in der Simulation sollte entweder schon besprochen worden sein oder eine Besprechung im Zuge dieser Aktivität erfolgen, dann Dauer anpassen.				

AktivitätsschabloneID:	E4S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren			Dauer:	ca. 3 - 4 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung einer Simulation eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung mind. drei Richtungsänderungen der Simulation des Bodenroboters notwendig sind.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm einer Simulation eines Bodenroboters und darf sich selbst eine Aufgabe zur Programmierung der Simulation des Bodenroboters stellen. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe mind. drei Richtungsänderungen programmiert werden müssen. Das Kind programmiert den Weg ein, TR unterstützt es dabei, falls erforderlich.				
Ergänzende Informationen:	Die Steuerung des Bodenroboters in der Simulation sollte entweder schon besprochen worden sein oder eine Besprechung im Zuge dieser Aktivität erfolgen, dann Dauer anpassen.				

8.7.2 Framework Perspektive 2: Aktivitätsschablonen nach Leitthemen

Nachdem in der Framework Perspektive 1 (siehe Kapitel 8.7.1) die Aktivitätsschablonen entlang der Inhaltsdimensionen des Frameworks vorgestellt wurden, folgt in der Framework Perspektive 2 eine Übersicht über die Zuordnung der Aktivitätsschablonen zu den Leitthemen und zur Erreichung von Lernergebnissen.

Eine Übersicht über die möglichen Lernergebnisse findet sich in Kapitel 8.2.1.2, 8.2.2.2 und 8.2.3.2.

Aktivitäts-schablone	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5
	Leitthema: Early Childhood Technology																									
C1S1	x				x	x	x																			
C2S1				x	x	x																				
C2S3	x	x	x		x																				x	
C2S4					x	x	x																			x

Aktivitäts- schablone	Leitthema: Computational Thinking																											
	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5		
C4S2	x	x				x		x	x			x	x			x							x	x			x	
C4S3		x			x	x			x															x		x	x	
D1S1	x			x	x																			x			x	
D1S2	x	x									x													x				
D2S3		x																					x					
D3S2	x		x																							x		
D3S3					x						x																	
E2S1	x								x								x	x						x			x	
E3S4						x	x									x	x					x					x	
Leitthema: Computational Thinking																												
A1S3								x		x			x					x									x	
A2S4							x		x			x	x		x	x	x	x						x	x			
A3S1								x		x			x		x		x	x	x			x						
A3S2								x		x	x	x	x		x		x	x	x			x						
A3S3								x		x			x		x		x	x	x			x						
A3S4								x		x			x	x	x		x	x	x	x	x							
A4S1								x		x			x	x			x	x	x			x					x	
A4S2								x		x			x	x			x	x	x			x					x	
A4S3								x		x			x	x			x	x	x			x					x	
A4S4								x		x			x	x			x	x	x			x					x	
A5S4								x		x			x		x		x	x	x			x						
A6S3						x		x	x	x																	x	
A6S4						x	x	x	x				x					x		x							x	
B1S4	x								x					x	x			x	x	x	x						x	
B2S2												x	x									x					x	
B2S3								x	x	x				x					x	x								
B3S2							x			x			x	x	x				x	x	x	x					x	
B3S3							x			x			x	x	x				x	x	x	x					x	
B3S4							x			x			x	x	x				x	x	x	x					x	
B4S1							x							x													x	
B4S2							x		x			x					x										x	
B4S3								x	x																		x	
B4S4							x	x		x															x		x	
E4S1							x	x		x						x	x	x	x									
E4S2							x	x		x						x	x	x	x									
E4S4							x	x		x	x					x	x	x	x								x	
Leitthema: Child-Centered-Design																												
A1S1						x																				x	x	
A1S2								x		x	x	x	x	x		x	x							x	x	x	x	
A1S4						x			x			x	x						x					x		x	x	
A2S3								x	x	x		x	x		x	x	x						x	x	x			
C1S2					x		x																	x			x	
C1S3																							x		x	x	x	
C3S2								x															x			x	x	
C3S4																								x			x	
D2S4	x						x		x									x									x	x
E1S4							x										x	x									x	x
E3S3							x	x					x	x			x	x	x	x							x	x

Tabelle 10: Aktivitätsschablonen nach Leitthema und Lernergebnissen

Die angeführten Lernergebnisse werden ebenso im Zusammenhang mit früherer Technikbildung in Informatik bei der Evaluierung der Hardware Usability (siehe Kapitel 9.1.7.3), bei der Evaluierung der Software Usability (siehe Kapitel 9.2.7.3) sowie durch eine Teilnahme am adaptierten Child-Centered-Design-Prozess (siehe Kapitel 10.5) gefördert.

8.7.3 Framework Perspektive 3: Aktivitätsschablonen nach Lernergebnissen

Die dritte Framework Perspektive listet die Aktivitätsschablonen zugeordnet zu den Lernergebnissen und geordnet nach den Inhaltsdimensionen auf.

Aktivitäts-schablone	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5		
Inhaltsdimension A: Programmierumgebung Hardware																												
A1S1						x																		x	x			
A1S2								x		x	x	x	x	x		x	x						x	x	x	x		
A1S3								x		x			x					x								x		
A1S4						x			x			x	x							x				x		x	x	
A2S3								x	x	x		x	x		x	x	x					x	x	x				
A2S4							x		x			x	x		x	x	x	x						x	x			
A3S1								x		x			x		x		x	x	x			x						
A3S2								x		x	x	x	x		x		x	x	x			x						
A3S3								x		x			x		x		x	x	x			x						
A3S4								x		x			x	x	x		x	x	x	x	x							
A4S1								x		x			x	x			x	x	x			x					x	
A4S2								x		x			x	x			x	x	x			x					x	
A4S3								x		x			x	x			x	x	x			x					x	
A4S4								x		x			x	x			x	x	x			x					x	
A5S4								x		x			x		x		x	x	x			x						
A6S3						x		x	x	x																	x	
A6S4							x	x		x			x					x		x						x		
Inhaltsdimension B: Informatische Lernumgebung technikfrei																												
B1S4	x								x					x	x			x	x	x	x						x	
B2S2												x	x					x		x							x	
B2S3								x	x	x				x				x	x									
B3S2							x			x			x	x	x			x	x	x	x						x	
B3S3							x			x			x	x	x			x	x	x	x						x	
B3S4							x			x			x	x	x			x	x	x	x						x	
B4S1							x								x												x	
B4S2							x		x			x					x		x								x	
B4S3								x	x																		x	
B4S4							x	x		x														x		x	x	
Inhaltsdimension C: Wissenschaftliche Herangehensweisen																												
C1S1	x					x	x	x																				
C1S2						x		x															x				x	
C1S3																					x			x	x		x	
C2S1					x	x	x																					
C2S3	x	x	x			x																					x	
C2S4						x	x	x																			x	
C3S2								x													x				x	x		
C3S4																							x				x	
C4S2	x	x				x		x	x			x	x				x						x				x	
C4S3		x				x	x			x														x			x	
Inhaltsdimension D: Technische Alltagserfahrungen																												
D1S1	x				x	x																					x	
D1S2	x	x												x													x	
D2S3		x																									x	
D2S4	x							x		x											x						x	
D3S2	x																										x	
D3S3																												
Inhaltsdimension E: Programmierumgebung Software																												
E1S4																											x	x

Aktivitäts- schablone	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5
E2S1	x								x							x	x						x			x
E3S3						x	x					x	x			x	x	x	x						x	x
E3S4						x	x					x				x	x				x					x
E4S1							x	x		x						x	x	x	x							
E4S2							x	x		x						x	x	x	x							
E4S4							x	x		x	x		x			x	x	x	x						x	

Tabelle 11: Aktivitätsschablonen nach Lernergebnissen und Inhaltsdimensionen

8.7.4 Framework Perspektive 4: Aktivitätsschablonen nach Tätigkeiten

Die Framework-Perspektive 4 zeigt eine Übersicht der Aktivitätsschablonen geordnet nach den enthaltenen Tätigkeiten wiederum im Zusammenhang mit den Lernergebnissen.

Aktivitäts- schablone	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5	
Bewegen																											
B1S4	x								x					x	x			x	x	x	x					x	
B2S2												x	x					x	x	x	x						x
B3S2							x			x			x	x	x			x	x	x	x					x	
B3S3							x			x			x	x	x			x	x	x	x					x	
B3S4							x			x			x	x	x			x	x	x	x					x	
B4S1							x							x												x	
Demonstrieren																											
A2S3								x	x	x		x	x		x	x	x						x	x	x		
A2S4							x			x		x	x		x	x	x	x						x	x		
B2S3								x	x	x				x					x	x							
D1S1	x			x	x																			x		x	
D1S2	x	x									x													x			
E2S1	x								x							x	x							x		x	
Dokumentieren																											
C2S1				x	x	x																					
C2S3	x	x	x		x																					x	
C2S4					x	x	x																			x	
D3S3					x						x																
Erforschen																											
A1S1						x																		x	x		
A1S2								x		x	x	x	x	x		x	x						x	x	x	x	
D3S2	x		x				x																			x	
E1S4							x									x	x									x	
E3S3						x	x					x	x			x	x	x	x							x	
E3S4						x	x					x				x	x					x				x	
Fragen entwickelnd unterrichten																											
A6S3						x		x	x	x																	
B4S3								x	x																	x	
C1S1	x				x	x	x																				
D2S3		x					x																x			x	
D2S4	x						x		x									x								x	
Gestalten																											
B4S4							x	x		x														x		x	
C1S2					x		x																x			x	
C3S2							x													x					x	x	
C3S4																					x					x	

Aktivitäts- schablone	LE1.1	LE1.2	LE1.3	LE1.4	LE1.5	LE1.6	LE1.7	LE2.1	LE2.2	LE2.3	LE2.4	LE2.5	LE2.6	LE2.7	LE2.8	LE2.9	LE2.10	LE2.11	LE2.12	LE2.13	LE2.14	LE3.1	LE3.2	LE3.3	LE3.4	LE3.5	
Präsentieren																											
A1S4						x			x			x	x								x			x		x	x
C1S3																					x			x	x		x
Programmieren																											
A1S3								x		x			x				x										x
A3S1								x		x			x		x		x	x	x		x						
A3S2								x		x	x	x	x		x		x	x	x		x						
A3S3								x		x			x		x		x	x	x		x						
A3S4								x		x			x	x	x		x	x	x	x	x						
A4S1								x		x			x	x			x	x	x		x						x
A4S2								x		x			x	x			x	x	x		x						x
A4S3								x		x			x	x			x	x	x		x						x
A4S4								x		x			x	x			x	x	x		x						x
A5S4								x		x			x		x		x	x	x		x						
A6S4								x	x			x						x			x						x
E4S1								x	x							x	x	x	x								
E4S2								x	x							x	x	x	x								
E4S4								x	x							x	x	x	x								x
Wiederholen																											
B4S2								x		x			x				x		x								x
C4S2	x	x				x		x	x			x	x			x							x	x			x
C4S3		x				x	x			x															x		x

Tabelle 12: Aktivitätsschablonen nach Tätigkeiten und Lernergebnissen

8.8 Arbeiten mit dem Framework

Der folgende Abschnitt demonstriert die exemplarische Anwendung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Hierzu zeigt die Verfasserin der gegenständlichen Arbeit, wie die vier Perspektiven des Frameworks bei der Auswahl einer geeigneten Aktivitätsschablone unterstützen und wie anschließend eine beispielhafte Aufgabe aus der jeweils ausgewählten Aktivitätsschablone abgeleitet werden kann.

Bei allen angeführten Beispielen gilt, dass entweder eine Kindergartenpädagogin, ein Kindergartenpädagoge oder eine zum Kindergarten außenstehende Person in der Rolle der Trainerin oder des Trainers, kurz TR, agiert. Die Ableitung von Aufgaben folgt einem Top-Down-Ansatz, von einem abstrakten Leitgedanken ausgehend, über eine schon weniger abstrakte und tendenziell konkretere Leitfrage, hin zur Auswahl einer geeigneten Schablone und Ableitung einer an den eigenen Kontext angepassten Aufgabe.

8.8.1 Ableitung einer Aufgabe mit Perspektive 1 nach Inhaltsdimension

Leitgedanke: Ich möchte einen bestimmten Inhalt vermitteln.

Leitfrage: Welche Aktivitätsschablone soll ich wählen, wenn ich ohne Einsatz von Technik den teilnehmenden Kindern den Begriff „elementarer Verarbeitungsschritt“ erklären möchte?

Leitantwort: In der Inhaltsdimension B, „Informatische Lernumgebung technikfrei“, finden sich Aktivitätsschablonen zur frühen Technikbildung in Informatik ohne Einsatz von technischen Materialien. Der Aktivitätscluster B1, „Verbalisierte elementare Verarbeitungsschritte“, beinhaltet Aktivitätsschablonen, die helfen, den Kindern den Begriff „elementarer Verarbeitungsschritt“ zu vermitteln. Es kann zum Beispiel die Aktivitätsschablone mit der ID B1S4 gewählt werden.

Ausgewählte Aktivitätsschablone:

AktivitätsschabloneID:	B1S4	Schwierigkeitsgrad:	4	Leitthema	Computational Thinking
Aktivitätscluster:	B1) Verbalisierte elementare Verarbeitungsschritte			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen			Dauer:	ca. 15 - 20 Minuten
Aktivitätsschablone:	<i>Ein Kind spielt Roboter und wird durch verbale Befehlseingaben durch ein zweites Kind „gesteuert“.</i>				
Material:	Es werden keine besonderen Materialien benötigt.				
Setting:	Die Kinder sollen sich in 2er-Teams zusammenfinden und führen die Aufgabe in den 2er-Teams nacheinander in der Großgruppe durch. Ein Kind des 2er-Teams spielt Roboter und wird durch verbale Befehlseingaben vom anderen Kind gesteuert. Das Kind, das den Roboter spielt, führt nur genau das, was angesagt wird, ohne Interpretation und Ergänzungen aus. Die gerade nicht ins Roboter-Spiel involvierten Kinder sitzen im Sitz-Halbkreis und unterstützen mit Hilfestellungen, falls erforderlich.				
Ergänzende Informationen:	Sollte von TR eindeutig exemplarisch vorgezeigt werden um u.a. die Vorgehensweise und notwendige Genauigkeit beim Formulieren und Ausführen der Befehle zu verdeutlichen. Bei der Erklärung der Aufgabe darauf eingehen, dass ein Roboter programmiert wird und nur genau die Befehle, die eingespeichert werden, exakt ausführt ohne Ergänzungen und Interpretationen vorzunehmen. Unterschied zum Menschen, dessen Gehirn die einzelnen Körperfunktionen steuert, erklären – Bogen spannen zu dieser Aufgabe, wo im 2er-Team das Kind, das die Anweisungen gibt, das andere Kind, das „Roboter“ spielt, bezüglich der Bewegungen programmiert.				

Beispiel einer abgeleiteten Aufgabe: Immer zwei Kinder bilden eine Kleingruppe, in der sich die Kinder abwechselnd mit Sprachbefehlen in Form von elementaren Verarbeitungsschritten „steuern“ sollen.

Hinweise zur Durchführung der Aufgabe: Die Aufgabe kann im Rahmen einer Geschichte (story telling) genutzt werden. Zu Beginn erörtert TR die Geschichte – Zum Beispiel sollen die Roboter nach vermissten Gegenständen, etwa einem Ball, im Raum suchen.

Bevor die Roboter sich jedoch auf die Suche machen, muss TR gemeinsam mit den Kindern besprechen, was ein elementarer Verarbeitungsschritt sein soll. „Gehen“ ist zum Beispiel kein elementarer Verarbeitungsschritt, da „Gehen“ aus einzelnen „Schritten“ besteht. Jedoch kann „ein Schritt vorwärts“ im Kontext dieser Aufgabe durchaus als elementarer Verarbeitungsschritt betrachtet werden.

Nach dieser Erörterung sollte TR noch die „Spielregeln“ festlegen. Die Kinder sollten darauf achten, dass diejenigen, die in der Rolle als Roboter unterwegs sind, nicht mehr machen, als ihnen gesagt wird und sie nach jedem einzelnen Befehl wieder aufhören – also ihre Bewegungen stoppen. Die Kinder, die die Sprachsteuerung des Roboters mittels elementarer Verarbeitungsschritte übernehmen, sollten immer nur einen Befehl nennen und die Ausführung des Befehls abwarten, bevor ein weiterer Befehl erfolgen kann.

Bei dieser Übung können entweder alle Kleingruppen gleichzeitig aktiv sein, oder jede Gruppe nacheinander. Bei der ersten Durchführung dieser Aufgabe ist eine Durchführung hintereinander – eine Kleingruppe nach der anderen – zu empfehlen, damit TR die Gelegenheit hat, auf die Einhaltung der „Spielregeln“ und auf den Einsatz von elementaren Verarbeitungsschritten zu achten.

Die Aufgabe kann über die Dauer ihrer Durchführung erweitert werden, so dass die Kinder, die die Sprachbefehle geben, die Kinder, die als Roboter agieren, tatsächlich zum Suchen von vermissten Gegenständen einsetzen und diesen Gegenstand, zum Beispiel einen Ball, nach dem Finden aufheben und mitnehmen, zurück zur Ausgangsposition.

TR sollte während der Durchführung der Aufgabe darauf achten, dass alle Kinder zumindest einmal Sprachbefehle gegeben haben und alle Kinder zumindest einmal in der Rolle eines Roboters aktiv waren.

Zum Abschluss kann TR mit den Kindern kurz über die Aufgabe reflektieren und ggf. nochmals auf den Begriff des elementaren Verarbeitungsschritts eingehen.

8.8.2 Ableitung einer Aufgabe mit Perspektive 2 nach Leitthema

Leitgedanke: Ich möchte Inhalte zu einem bestimmten Leitthema vermitteln.

Leitfrage: Welche Aktivitätsschablone soll ich wählen, wenn ich Inhalte zum Thema „Child-Centered-Design“ vermitteln möchte und mich hierbei auf eine wissenschaftliche Herangehensweise sowie die Erstellung von Lernmaterialien zum unmittelbaren Einsatz konzentrieren möchte?

Leitantwort: Zum Leitthema „Child-Centered-Design“ bietet zum Beispiel die Inhaltsdimension C, „Wissenschaftliche Herangehensweisen“ Aktivitätsschablonen an. Im Aktivitätscluster „Konzeption und Erstellung Lernmaterial“ findet sich zum Beispiel die Aktivitätsschablone mit der ID C3S2, die sich mit den gefragten Kriterien beschäftigt.

Ausgewählte Aktivitätsschablone:

AktivitätsschablonenID:	C3S2	Schwierigkeitsgrad:	2	Leitthema	Child-Centered-Design
Aktivitätscluster:	C3) Konzeption und Erstellung Lernmaterial			Sozialform:	Kleingruppe
Aufgabenart:	Gestalten			Dauer:	ca. 25 - 30 Minuten
Aktivitätsschablone:	<i>Die Kinder sollen Arbeitsmatten zur Nutzung mit dem gewählten Bodenroboter frei gestalten.</i>				
Material:	Blätter mit einem vorgezeichneten Raster in Schrittweite des Bodenroboters – je Kleingruppe von drei Kindern mindestens ein Blatt (z.B. 5 mal 4 Kästchen auf Flipchart- oder Backpapier), Farbstiftsets, Spitzer und Radiergummi für jede Kleingruppe, wiederablösbares Klebeband um die Mattenrohlinge am Boden festzukleben				
Setting:	Die Kinder sollen die Mattenrohlinge in Kleingruppen mit eigenen Ideen gestalten. Welche Welt soll der Bodenroboter auf der Arbeitsmatte erkunden können?				
Ergänzende Informationen:	Ausreichender Platz rund um die jeweilige Zeichnung (am Boden - die Kinder malen gern am Boden auf Polstern rund um die Matte sitzend) wird benötigt, um zu verhindern, dass die Kinder beim Holen von Stiften, Wechseln der Mattenseite, etc. auf die Matte steigen oder gar die Ecken umknicken / einreißen etc.				

Beispiel einer abgeleiteten Aufgabe: Die Kinder sollen eine Arbeitsmatte in Form einer Baustelle gestalten, auf welcher der Constructa-Bot fahren kann.

Hinweise zur Durchführung der Aufgabe: Der Constructa-Bot ist von seiner Funktion gleich gestaltet wie der Bee-Bot, jedoch ist sein Aussehen einem Baufahrzeug, im Speziellen einem Lastwagen mit einem Greifarm vorne, nachempfunden.

Die Kinder sollen daher auf einer Matte mit 8x6 quadratischen Kästchen, zu einer Kantenlänge von je 15 cm, eine Szene bzw. Geschichte gestalten, auf welcher der Constructa-Bot fahren kann. Es empfiehlt sich hier das Zeichnen einer Baustelle und zum Beispiel eines Lagerhauses, aus dem Materialien geholt und zur Baustelle gebracht werden können.

Zur Gestaltung der Arbeitsmatte können sowohl Zeichenmaterialien wie auch Bastelmaterialien eingesetzt werden. So könnten zum Beispiel aus leeren Klopapierrollen Säulen aufgeklebt werden, welchen der Constructa-Bot ausweichen muss oder mit kleinen Schachteln Häuser aufgestellt werden. Der Phantasie (der Kinder) sind in dieser Aufgabe keine Grenzen gesetzt.

Die Arbeitsmatte sollte während der Erstellung mit Klebestreifen zum Beispiel am Boden befestigt werden, damit diese nicht verrutscht, wenn Kinder ihre Arbeitsposition rund um die Arbeitsmatte wechseln.

Im Rahmen der Durchführung der Aufgabe kann den Kindern zusätzlich erörtert werden, dass der Constructa-Bot über eine sog. Schrittweite von 15 cm verfügt. Das heißt, wann immer er einen Schritt vorwärts oder rückwärts fahren soll, fährt er 15 cm. Daher wird die Arbeitsmatte auch in 15cm-Quadrate unterteilt, damit die Kinder anschließend den Constructa-Bot von einem zu definierenden Startpunkt zu einem zu definierenden Endpunkt mit Hilfe ihrer selbst durchgeführten Programmierung fahren lassen können.

Die Kinder nehmen in dieser Aufgabe die Rolle der Gestalterinnen und Gestalter ein und entwickeln Lernmaterial, das sie sich selbst und anderen Kindern zur frühen Technikbildung in Informatik zur Verfügung stellen können.

8.8.3 Ableitung einer Aufgabe mit Perspektive 3 nach Lernergebnis

Leitgedanke: Ich möchte mit der abgeleiteten und eingesetzten Aufgabe ein bestimmtes Lernergebnis unterstützen.

Leitfrage: Welche Aktivitätsschablone soll ich wählen, um Kindern die Möglichkeit zu geben, (früher) erworbenes themenrelevantes Wissen abzurufen und zur Lösung neuer selbst gestellter Aufgabenstellungen im Themenbereich der Sequenzen in der Simulation des Bodenroboters anzuwenden?

Leitantwort: Das Lernergebnis (LE 1.7) „... (früher) erworbenes themenrelevantes Wissen abzurufen und zur Lösung neuer Aufgabenstellungen anzuwenden“ wird zum Beispiel von der Inhaltsdimension E „Programmierungsumgebung Software“ unterstützt. Mit dem Aktivitätscluster „Selbstgewähltes Problem lösen: Sequenzen“ haben die Kinder die Möglichkeit, das Lernziel mittels Aufgaben, abgeleitet aus der Schablone E4S1, zu erreichen.

Ausgewählte Aktivitätsschablone:

AktivitätsschablonenID:	E4S1	Schwierigkeitsgrad:	1	Leitthema	Computational Thinking
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren			Dauer:	ca. 2 Minuten
Aktivitätsschablone:	<i>Kind stellt sich selbst eine Aufgabe zur Programmierung einer Simulation eines Bodenroboters. TR achtet bei der Aufgabenfindung durch das Kind darauf, dass zur Lösung keine Richtungsänderung der Simulation des Bodenroboters notwendig ist.</i>				
Material:	Notebook mit installierter Simulation eines Bodenroboters, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind				
Setting:	Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm einer Simulation eines Bodenroboters und darf sich selbst eine Aufgabe zur Programmierung der Simulation des Bodenroboters stellen. TR unterstützt die Aufgabenfindung des Kindes, indem darauf geachtet wird, dass zur Lösung der Aufgabe keine Richtungsänderung programmiert werden muss. Das Kind programmiert den Weg ein, TR unterstützt es dabei, falls erforderlich.				
Ergänzende Informationen:	Die Steuerung des Bodenroboters in der Simulation sollte entweder schon besprochen worden sein oder eine Besprechung im Zuge dieser Aktivität erfolgen, dann Dauer anpassen.				

Beispiel einer abgeleiteten Aufgabe: Das Kind soll sich selbst eine Aufgabe in der Software Focus on Bee-Bot 1, als Simulation des eingesetzten Bodenroboters Bee-Bot, auswählen.

Das Kind möchte auf einer virtuellen Arbeitsmatte mit geometrischen Figuren vom roten Kreis zum grünen Quadrat fahren.

Hinweise zur Durchführung der Aufgabe: Die Lösung der vom Kind selbst gewählten Aufgabe lautet: der virtuelle Bee-Bot muss drei Schritte vorwärts fahren.

TR muss bei dieser Aufgabe darauf achten, dass das Kind sich nicht zu komplexe Aufgaben stellt – es sollten daher, gerade zu Beginn der Übungen mit der Software-Simulation, keine Richtungsänderungen in der Lösung enthalten sein. Weiters muss die Lösung der vom Kind selbst gewählten Aufgabe mindestens zwei elementare Verarbeitungsschritte enthalten, sonst wird keine Sequenz zur Lösung benötigt.

Das Kind sollte vorher schon mit dem realen Bee-Bot Problemstellungen selbständig gelöst haben und nun in dieser Aufgabe sein Vorwissen zur Bedienung des virtuellen Bodenroboters nutzen, um die Lösung zu erarbeiten.

Die Aufgabe kann in unzähligen Variationen ausgestaltet werden. TR muss hier zum einen auf die gewählte Aufgabenstellung achten und zum anderen auf die Rahmenbedingungen, so dass das jeweilige Kind ergonomisch mit der Software-Simulation des Bodenroboters

arbeiten kann. Allfällige Hilfestellungen von TR an das Kind sind eher in Form von Fragen anzubieten, als mit der Präsentation der Lösung.

8.8.4 Ableitung einer Aufgabe mit Perspektive 4 nach Tätigkeit

Leitgedanke: Ich möchte Inhalte mit einer bestimmten Tätigkeit unterstützt vermitteln.

Leitfrage: Welche Aktivitätsschablone soll ich wählen, wenn ich mittels „Dokumentieren“ als Tätigkeit die Exploration technischer Alltagsgeräte ermöglichen möchte?

Leitantwort: Die Inhaltsdimension D, „Technische Alltagserfahrungen“ bietet hierzu Aktivitätsschablonen an. Im Aktivitätscluster „Exploration technischer Alltagsgeräte“ kann mit der Tätigkeit „Dokumentieren“ und Aufgaben, abgeleitet aus der Aktivitätsschablone D3S3, Kindern nicht nur die Tätigkeit „Dokumentieren“ nähergebracht werden, sondern auch der Einsatz technischer Alltagsgeräte zum Zwecke der Dokumentation.

Ausgewählte Aktivitätsschablone:

AktivitätsschablonenID:	D3S3	Schwierigkeitsgrad:	3	Leitthema	Early Childhood Technology
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Dokumentieren			Dauer:	ca. 15 - 20 Minuten
Aktivitätsschablone:	<i>Die Kinder drucken mit Unterstützung durch TR die Fotos, die mit altersgerechten digitalen Fotoapparaten erstellt wurden, aus.</i>				
Material:	Fotodrucker, der ohne Verbindung zum Computer direkt von Speicherkarte oder durch Verbindung zur Kamera mit USB-Kabel Fotos drucken kann, ausreichend Fotopapier für diesen Drucker, vergewissern dass Patronen genügend gefüllt sind				
Setting:	Alle sitzen im Sitzkreis. Jedes Kind darf reihum ein Foto ausdrucken, das mit den altersgerechten digitalen Fotoapparaten vom Einsatz eines technischen Geräts mit Informatikbezug im Alltag gemacht wurde.				
Ergänzende Informationen:	Das Funktionsprinzip des verwendeten Druckers kann hier während des Ausdrucks erklärt werden (z.B. wenn das Foto immer wieder eingezogen wird und sich die Farben des Fotos langsam aufbauen).				

Beispiel einer abgeleiteten Aufgabe: Die Kinder sollen sich auf ihren altersgerechten digitalen Fotoapparaten ihr Lieblingsbild aussuchen. TR druckt anschließend gemeinsam mit den Kindern die jeweiligen Fotos mit einem Fotodrucker aus. Das jeweils ausgedruckte Foto wird zu Dokumentationszwecken aufbewahrt.

Hinweise zur Durchführung der Aufgabe: Bevor die Fotos ausgedruckt werden, sollte TR die grundlegende Funktionalität des eingesetzten Fotodruckers zielgruppengerecht erklären. Hier könnte zum Beispiel ein „Canon® SELPHY CP-1300“ Fotodrucker zum Einsatz kommen. Dieser Drucker arbeitet nach dem Prinzip des Thermosublimationsdrucks und bringt nacheinander das gesamte Bild in den drei Grundfarben (Yellow, Magenta und Cyan) der subtraktiven Farbmischung auf. Zuletzt wird das Foto noch mit einer Schutzschicht überzogen.

Der Prozess zum Drucken des Fotos könnte wie folgt aussehen:

- Kinder wählen sich jeweils ihr Lieblingsfoto auf ihren altersgerechten digitalen Fotoapparaten (zum Beispiel einem „Kidizoom Touch 5.0“) aus.
- TR schreibt sich die Nummer (Dateiname o.ä.) vom Foto auf, damit das Foto wiedergefunden werden kann.
- TR entnimmt die Speicherkarte und gibt diese in den Fotodrucker.
- TR erklärt den Kindern, wie man über das Display am Fotodrucker das gewünschte Foto findet.
- Kind wählt unterstützt von TR das gewünschte Foto aus.
- Kind druckt das gewünschte Foto aus und TR erklärt, was der Drucker macht.
- Die Kinder entnehmen nacheinander ihr Lieblingsfoto dem Fotodrucker und dürfen dieses aufhängen, in ihr Forschungsheft kleben oder mit nach Hause nehmen.

Während den Erklärungen sollte TR die Kinder immer wieder darauf aufmerksam machen, dass die Tastendrucke am Fotodrucker „Befehle“ sind, die dem Drucker „sagen, was er machen soll“. Weiters sollte TR hervorheben, dass der Drucker mit seinem Display, seinen Steuerungstasten und seinen Schnittstellen ein „Spezial-PC“ ist, der ähnlich zu einem, den Kindern vermutlich bekannten, herkömmlichen PC funktioniert, aber auf eine Aufgabe – das Drucken von Fotos – spezialisiert ist.

Im Rahmen der Aufgabe sollte TR regelmäßig darauf eingehen, dass das Erstellen von Fotos und das anschließende Ausdrucken selbiger eine Methode zur Dokumentation von Erlebnissen, Ereignissen oder zu beobachtenden Phänomenen darstellt.

9 Evaluierung von Hardware und Software

Die Evaluierung der Hardware Usability und der Software Usability verfolgt mehrere Ziele.

Zum einen gilt es herauszufinden, ob die zum Einsatz intendierten Hardware- und Softwareprodukte (der Bodenroboter, hier der Bee-Bot und die korrespondierende Softwaresimulation Focus on Bee-Bot 1 bzw. 2, siehe Kapitel 4.4.2 und Kapitel 4.3.1.3.3) für die Zielgruppe geeignet sind und über eine entsprechende Benutzerfreundlichkeit verfügen. Wie Markopoulos et al. (2008) in Kapitel 5.1.1.2 darlegen ist die Evaluierung der Usability von Hard- und Software mit 3- bis 6-jährigen Kindern gut möglich und wird daher in Folge im Rahmen der gegenständlichen Forschungsarbeit durchgeführt.

Zum anderen erfolgt der Einsatz der Evaluierungen der Hardware Usability und Software Usability als gezielte Aktivität im Child-Centered-Design, welches als Maßnahme zur frühen Technikbildung in Informatik im Rahmen der gegenständlichen Arbeit erforscht wird. Die Beiträge der beiden Evaluierungen zur frühen Technikbildung sind den jeweiligen Kapiteln (siehe Kapitel 9.1.7.3 und 9.2.7.3) zu entnehmen.

Darüber hinaus dient die Evaluierung der Hardware Usability und Software Usability der Beantwortung der korrespondierenden Forschungsfrage (siehe Kapitel 1.3.1.1).

Zu Beginn beschäftigt sich Kapitel 9.1 mit der Evaluierung der Hardware Usability und anschließend Kapitel 9.2 mit der Evaluierung der Software Usability.

9.1 Evaluierung Hardware Usability

Der Hardware Usability Test, welcher aus zwei Testteilen mit Gruppen unterschiedlichen Vorwissens besteht, bildet einen Teil der gesamten Usability Evaluierung des Bee-Bots und der Softwareprodukte Focus on Bee-Bot 1 bzw. 2.

Am Hardwaretest 1 (HWT1) nahmen zehn Kinder aus dem Kindergarten D teil, um die Usability des Bee-Bots zu evaluieren. Diese Kinder hatten zuvor nach Angaben der Kinder und der betreuenden Pädagoginnen und Pädagogen noch nie Kontakt mit dem Bee-Bot, weder in Form der Hardware noch in Form der Software.

Am Hardwaretest 2 (HWT2) nahmen neun Kinder aus dem Kindergarten E teil, um die Usability des Bee-Bots zu evaluieren. Diese Kinder hatten am Tag vor dem Hardwaretest 2 im Rahmen des Software Usability Tests die Software Focus on Bee-Bot 1 bzw. Focus on Bee-Bot 2 kennengelernt. Sie verfügten daher bereits über erste Kenntnisse zum virtuellen Bee-Bot. Im Rahmen des Hardware Usability Tests war folglich auch ein möglicher Transfer des Wissens der Kinder, welches sie im Rahmen des Software Usability Tests erworben hatten, auf die Bedienung des Hardware Bee-Bots von Interesse.

Das Kapitel zur Evaluierung der Hardware Usability beginnt mit einer Vorstellung der Funktionsweise des Bee-Bots (Kapitel 9.1.1). Anschließend werden die Rahmenbedingung der Evaluation (Kapitel 9.1.2), die eingesetzten Methoden und Vorgehensweisen (Kapitel 9.1.3) sowie der Leitfaden zur zusammenfassenden Transkription (Kapitel 9.1.4) angeführt. Kapitel 9.1.5 stellt das Kategoriensystem zur Kodierung vor. In Kapitel 9.1.6 werden die Ergebnisse aus der Kodierung präsentiert und in Kapitel 9.1.7 interpretiert sowie zusammengefasst.

9.1.1 Funktionsweise Bee-Bot

Der Bee-Bot ist ein einfacher Bodenroboter der Firma TTS Group Ltd, der in Form einer Biene, die fahren kann, gestaltet ist.



Abbildung 22: Bee-Bot von vorne und hinten, Quelle: TTS Group, <http://www.tts-group.co.uk/bee-bot-rechargeable-floor-robot/1001794.html>, 20.07.2016

Die Funktionsweise des Bee-Bots ist schematisch als einfacher Zustandsautomat in Abbildung 23 dargestellt. Der Bee-Bot kann bis zu 40 Befehle speichern, die entweder einen Vorwärts- oder Rückwärts-Schritt, eine Links- oder Rechts-Drehung, jeweils um 90 Grad, oder eine Pause darstellen. Die Befehle werden nacheinander per Tastendruck am Rücken des Bee-Bots eingegeben und nach dem Prinzip FIFO – First In First Out vom Bee-Bot abgearbeitet. Wenn die Befehlsabarbeitung gestartet werden soll, dann ist die Taste „GO“ zu betätigen. Soll die Abarbeitung der Befehle gestoppt werden, ist wiederum „GO“ zu drücken. Ein erneutes Drücken von „GO“ lässt den Bee-Bot die gespeicherten Befehle wieder von Anfang an abarbeiten. Um den Speicher zu löschen, ist die Taste „Clear“ zu betätigen. Dann wird der Speicher ohne Rückfrage gelöscht.

Um mit dem Bee-Bot eine Links- oder Rechts-Drehung zu vollziehen, genügt ein Druck auf die Taste Links oder Rechts. Hier dreht sich der Bee-Bot um 90 Grad nach links oder rechts. Um einen Schritt nach links oder rechts zu gehen, ist zuerst die Taste Links oder Rechts zu betätigen, gefolgt von einem Tastendruck auf Vorwärts. Erst dann fährt der Bee-Bot tatsächlich einen Schritt nach links oder nach rechts.

Auf der Unterseite des Bee-Bots finden sich zwei Schiebeschalter – einer zum Ein- und Ausschalten des Bee-Bots und einer zum Ein- und Ausschalten der Soundeffekte. Daneben befindet sich, in der Version des Bee-Bots, die die Autorin im Rahmen der gegenständlichen Forschungsarbeit einsetzt, das Batteriefach zur Stromversorgung des Bee-Bots, welches mit einem Drehverschluss gesichert ist, damit Kinder nicht allzu einfach die Batterien entfernen können.

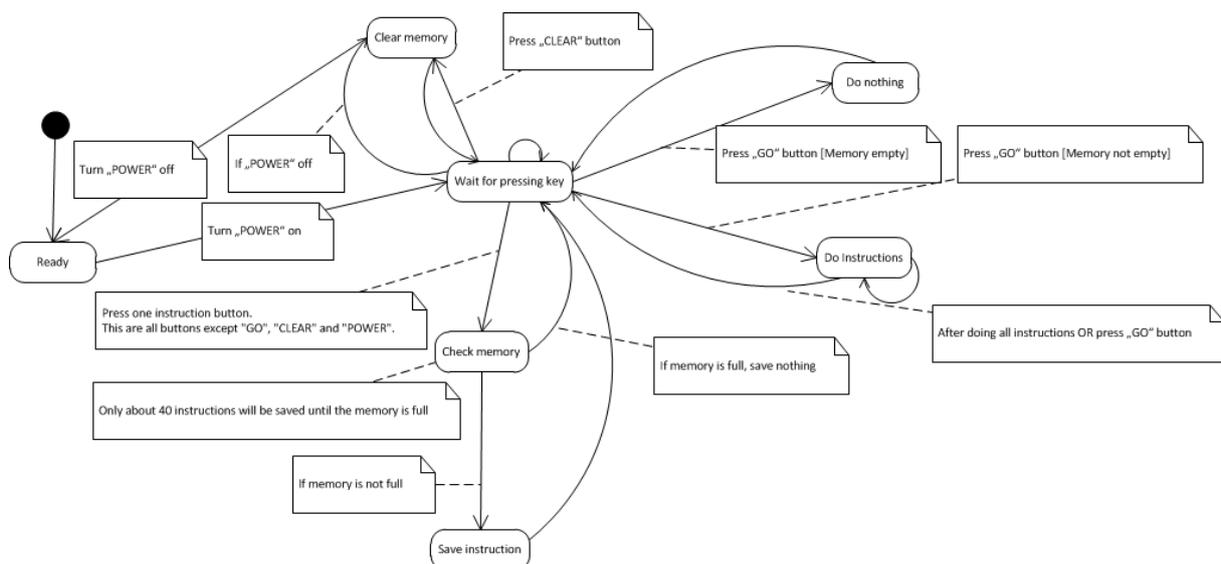


Abbildung 23: Einfacher Zustandsautomat zum Bee-Bot, übernommen aus (Stöckelmayr, Tesar & Hofmann, 2011, S. 186)

9.1.2 Beschreibung des Tests

9.1.2.1 Testziel

Ziel des Tests ist es herauszufinden, wie intuitiv der Bee-Bot für Kinder im Alter von 3 bis 6 Jahren gestaltet ist, so dass diese ihn ohne Anleitungen bzw. Instruktionen bedienen können. Darüber hinaus soll festgestellt werden, wie Kinder mit einfachen Problemstellungen, z.B. der Bee-Bot ist nicht eingeschaltet, umgehen können. Darüber hinaus soll die Evaluierung der Hardware Usability einen Beitrag zur frühen Technikbildung in Informatik (siehe Kapitel 9.1.7.3) leisten.

9.1.2.2 Kurzbeschreibung des Settings

Der Hardware Usability Tests fand in den Kindergärten in folgendem Rahmen statt: Die Autorin führte die Tests in einer für die Kinder vertrauten Umgebung, in einem Raum des jeweiligen Kindergartens, durch. Dazu konfrontierte die Autorin in der Rolle als Testleiterin (TL) jeweils ein Kind (K) mit dem Bee-Bot. Sowohl das Kind als auch die Testleitung nahmen am Fußboden auf Polstern Platz. Dies ist eine den Kindern sehr vertraute Art und Weise, sich mit (neuen) Dingen im Kontext des Kindergartens zu beschäftigen. Das Unterstützungspersonal zeichnete die Hardware Usability Tests vollständig mit Hilfe einer Videokamera auf und setzte ergänzend ein Tonaufnahmegerät ein, um ggf. die Verständlichkeit der Tonaufzeichnung zu verbessern bzw. als Sicherungslösung, sollte die Videokamera ausfallen. Darüber hinaus waren keine weiteren Personen an dem Test beteiligt.



Abbildung 24: Testsetting Hardware Usability Test

9.1.2.3 Demografische Daten

Die Testgruppen zum HWT1 und HWT2 setzten sich wie folgt aus Kindergartenkindern zusammen: Die Testgruppe des HWT1 aus Kindergarten D bestand aus fünf Mädchen und fünf Jungen im Alter von 4 bis 6 Jahren. Die Testgruppe des HWT2 aus Kindergarten E setzte sich aus vier Mädchen und fünf Jungen im Alter von 3 bis 6 Jahren zusammen.

	weiblich	männlich
HWT1	5	5
3 Jahre	0	0
4 Jahre	0	2
5 Jahre	1	1
6 Jahre	4 ³⁴	2

³⁴ Ein Mädchen war zum Zeitpunkt des Tests noch nicht 6 Jahre alt, da sie jedoch ebenso kurz vor der Einschulung stand, wie auch die anderen 6-jährigen Kinder, wurde sie der Gruppe der Schulanfängerinnen und -anfänger im Alter von 6 Jahren zugeordnet.

	weiblich	männlich
HWT2	4	5
3 Jahre	0	1
4 Jahre	1	1
5 Jahre	3	0
6 Jahre	0	3

Tabelle 13: Verteilung Alter und Geschlecht der Probandinnen und Probanden

9.1.3 Eingesetzte Methoden und Vorgehensweise

Die Verfasserin der gegenständlichen Arbeit setzte für den Hardware Usability Test einen halbstandardisierten Cognitive Walkthrough ein. Hierbei sollten zum einen die Kinder frei durch Ausprobieren und Experimentieren den Bee-Bot erkunden, andererseits sollte durch wenige Leitfragen gewährleistet werden, dass alle Aspekte des Bee-Bots evaluiert werden können. Besonderes Augenmerk legte die Testleitung darauf, dass die Themen Aussehen des Bee-Bots, Aktionen des Bee-Bots, Bedienung des Bee-Bots sowie Unterschiede zwischen Hardware und Software im Zuge des Tests vorkamen. Die Testleiterin führte die Kinder durch eine kindgerechte Interaktion und Gesprächsführung an diese Themen heran und motivierte sie zu Aussagen.

Für die anschließende Auswertung der Tests und ein Eingehen auf die zahlreichen Aktionen und Details eines solchen Tests mit Kindern erfolgte, wie schon erwähnt, eine konsequente Aufzeichnung der Tests mit Videokameras. Die Verfasserin der gegenständlichen Arbeit kombinierte als qualitative Methode die Videographie (siehe Kapitel 7.1.1.4) mit der qualitativen Inhaltsanalyse nach Mayring (siehe Kapitel 7.2). Durch Transkription machte die Autorin das Videomaterial lesbar (siehe Kapitel 9.1.4) und konnte das Ausgangsmaterial in weiterer Folge zusammenfassen (siehe Kapitel 7.2.3.1). Zur Auswertung des Materials bildete die Verfasserin der gegenständlichen Arbeit anschließend qualitative Kategorien, die das Auftreten spezieller Ereignisse und Aspekte repräsentieren. Ein von der Autorin erstellter Leitfaden zur Kodierung dient der Objektivierung derselben (siehe Kapitel 7.2.2). Abschließend erfolgte eine Ermittlung der Ergebnisse (siehe Kapitel 9.1.6).

Die Hardware Usability Tests fanden pro Kindergarten jeweils an einem Tag für alle Probandinnen und Probanden hintereinander statt. Ein Test dauerte zwischen 1:58 und 6:03 Minuten. Die Testleiterin verzichtete in Anwesenheit der Kinder weitestgehend auf das Wort „Test“, um unter allen Umständen zu vermeiden, dass für die Kinder eine Art Prüfungssituation entstehen könnte.

9.1.4 Leitfaden zur zusammenfassenden Transkription

Es erfolgt eine digitale Aufzeichnung der Tests mittels Videokamera, sodass sich die Testleitung während der Tests vollends auf den jeweiligen Test konzentrieren kann und nicht parallel auf die Dokumentation der Testergebnisse achten muss. Basierend auf den Aufzeichnungen erstellt die Autorin im Anschluss an die Tests Transkriptionen derselben: für jeden Test eine eigene Transkriptionsdatei, die zu Beginn das Datum des Tests, ein Pseudonym des Kindes, den Dateinamen des Videos und den anonymisierten Ort des Tests enthält.

Durch die Transkription beschränkt die Verfasserin der gegenständlichen Arbeit den Inhalt des Ausgangsmaterials auf das Wesentliche (siehe Kapitel 7.2.3.1), Irrelevantes wird weggelassen oder durch Verallgemeinerungen wie „Unterbrechung“ ersetzt. Als Wesentliches klassifiziert die Autorin alles, was die Fragestellung, Aktionen vom jeweiligen Kind und von der Testleitung sowie die Interaktionen zwischen ihnen und dem Bee-Bot betrifft. Alles, bei dem der Bee-Bot im Zentrum des Geschehens steht, erfasst die Verfasserin der gegenständlichen Arbeit so schriftlich und zusätzlich zum Videomaterial.

Die Autorin transkribiert Aussagen vom jeweiligen Kind und der Testleitung wortwörtlich, erfasst die Aktionen und Beobachtungen der Szenen durch Paraphrasierungen und erstellt zur Dokumentation pro Test folgende Tabelle (siehe Abbildung 25).

Minute	Aussage Kind	Aussage TL	Beobachtung	Aktion Kind	Aktion TL
22:35			Begrüßung, K nimmt auf dem Sitzpolster Platz und dreht sich in die richtige Richtung.		
22:44		Was ist denn das?		K schaut auf den Bee-Bot.	TL hält den Bee-Bot in der Hand und zeigt ihn K.
	Eine Biene.			K schaut zu TL und auf den Bee-Bot.	TL stellt den Bee-Bot auf den

Abbildung 25: Dokumentationsschema zur Transkription

Die Spalte Minute dient dem Vermerk der Zeitangabe der betreffenden Stelle im Video, um ein einfaches Auffinden derselben im Ausgangsmaterial zu ermöglichen. Die Zeitangabe erfolgt immer bei wesentlichen Ereignissen oder beim Wechsel zwischen einer wortwörtlichen Transkription und einer zusammenfassenden Beschreibung. In der Spalte „Aussage Kind“ sind die wortwörtlichen Meldungen des jeweiligen Kindes, in der Spalte „Aussage TL“ sind die wortwörtlichen Meldungen der Testleitung erfasst. In der Spalte „Beobachtung“ ist eine Zusammenfassung des Geschehens beschrieben. Zum Erfassen spezieller Handlungen und Aktionen dient für das Kind die Spalte „Aktion Kind“, für die Testleitung die Spalte „Aktion TL“.

9.1.5 Kategoriensystem zur Kodierung

Da es sich beim Hardware Usability Test um ein halbstandardisiertes Evaluationsverfahren handelt (siehe Kapitel 9.1.3), kennzeichnet die Verfasserin der gegenständlichen Arbeit in einem ersten Durchgang alle Fragen der Testleitung an das jeweilige Kind in den Transkripten mit einem „F“. Dies dient zum einen der leichteren Identifikation der wesentlichen Testinhalte und zum anderen in weiterer Folge als Unterstützung zur Kategorienfindung.

Das Ableiten der grundlegenden Kategorien zur Kodierung der Tests erfolgt deduktiv (siehe Kapitel 7.2.3.1). Ausgangspunkt bilden die grundlegenden Funktionalitäten und Eigenschaften des Bee-Bots und auch die Erkenntnisse aus der Vorstudie. Daraus entsteht das Kategoriensystem für den Hardware Usability Test in der Basisversion, Version 0.

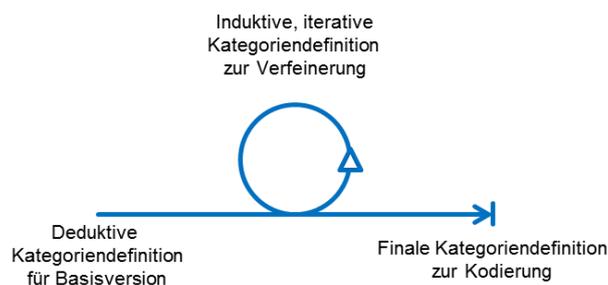


Abbildung 26: Vorgehen zur Definition des Kategoriensystems

Nach der Kodierung des ersten Hardware Usability Tests, auf Basis der Transkription des Tests und unter Einsatz des Kategoriensystems in der Basisversion, Version 0, erfolgte eine induktive Überarbeitung bzw. Ergänzung des Kategoriensystems durch die Autorin, da weitere interessante Aspekte aus der Transkription ersichtlich wurden. Es entstand das Kategoriensystem in der Version 1.

Das induktive Vorgehen zur Verfeinerung der Kategorien führte nach der Kodierung des zweiten Tests, auf Basis des Kategoriensystems in der Version 1 zu einer weiteren Ergänzung des Kategoriensystems. Es folgte Version 2 des Kategoriensystems.

Nach der Kodierung des dritten Tests mittels des Kategoriensystems in der Version 2 kam es erneut zur Ergänzung des Kategoriensystems, welches dann in Version 3 vorlag. Nach der

Kodierung von 13 Tests wurde das Kategoriensystem zum letzten Mal überarbeitet – es kam zur Version 4 des Kategoriensystems, welches als finales Kategoriensystem vorliegt und zur Kodierung der Hardware Usability Tests zum Einsatz kam.

Das gefestigte und für die Kodierung aller Hardware Usability Tests eingesetzte Kategoriensystem, welches die Autorin um eindeutige Identifizierungsnummern für die Kategorien ergänzte, sieht wie folgt aus:

HW01	Physikalische Eigenschaften des Bee-Bots
HW0101	Kind benennt Batteriefach
HW0102	Kind erkennt notwendige Stromversorgung
HW0103	Kind erwähnt Motor
HW0104	Kind benennt Bee-Bot als Roboter
HW0105	Kind benennt Bee-Bot als Biene
HW0106	Kind vermutet, dass der Bee-Bot fahren kann
HW02	Aktionen des Bee-Bots
HW0201	Kind thematisiert Augenblinken des Bee-Bots
HW0202	Kind bringt Augenblinken in Zusammenhang mit einem Tastendruck
HW0203	Kind benennt das Augenblinken des Bee-Bots als seine einzige Funktionalität bei Tastendruck
HW03	Bedienung des Bee-Bots
HW0301	Kind drückt als erstes folgende Taste
HW0302	Kind kann den Bee-Bot ohne Hilfe einschalten
HW0303	Kind kann mit Hinweisen den Bee-Bot einschalten
HW0304	Kind schiebt den Bee-Bot
HW0305	Kind kann die Funktion der Pfeil- bzw. Richtungstasten am Rücken des Bee-Bots erklären
HW0306	Kind kann die Funktion der GO-Taste am Rücken des Bee-Bots erklären
HW0307	Kind bringt Bee-Bot zum Fahren
HW0308	Kind gibt eine geeignete Befehls-Reihenfolge (Richtung, GO) bei eingeschaltetem Bee-Bot ein
HW0309	Kind kann die GO-Taste bewusst sinngemäß verwenden
HW0310	Kind kann die Richtungstasten bewusst sinngemäß verwenden
HW0311	Kind drückt während der Fahrt des Bee-Bots Tasten, um eine Richtungsänderung zu bewirken
HW0312	Die Art der Ausführung des Pause-Befehls führt zu Irritation bei Kind
HW0313	Kind drückt Clear- / Pause-Taste und anschließend die GO-Taste
HW0314	Kind gibt eine geeignete Befehls-Reihenfolge (Richtung, GO) bei ausgeschaltetem Bee-Bot ein
HW04	Herangehensweisen und Eigenschaften des Kindes
HW0401	Kind zeigt keine Berührungsängste mit dem Bee-Bot
HW0402	Kind hat Angst vor dem Bee-Bot (z.B. weil eine Biene stechen kann, ...)
HW0403	Kind spricht mit Bee-Bot
HW0404	Kind ist experimentierfreudig
HW0405	Kind ist zurückhaltend, eher verschlossen
HW0406	Kind weicht von sich aus dem Bee-Bot während der Fahrt aus, um dessen Fahrt nicht zu behindern
HW0407	Kind interagiert mit TL gleichberechtigt und aktiv, mit keinen bis wenigen Aufforderungen
HW0408	Der Grad an Interaktion des Kinds mit TL nimmt im Verlauf des Tests merklich zu

HW05	Interesse an Bee-Bot (nur eine Antwort)
HW0501	Kind zeigt Interesse an Bee-Bot
HW0502	Kind zeigt hohes Interesse an Bee-Bot
HW06	Analogien zur Software Focus on Bee-Bot 1 bzw. 2
HW0601	Kind erkennt Bee-Bot als Spielfigur der Software Focus on Bee-Bot wieder
HW0602	Kind sieht / thematisiert Analogie in der Bedienung zwischen realem und virtuellem Bee-Bot
HW0603	Kind kann Clear benennen / zeigen
HW0604	Kind kann die Funktionalität Clear sinngemäß benutzen
HW0605	Kind erkennt und benennt Unterschiede zwischen Bee-Bot und der Software Focus on Bee-Bot
HW0606	Kind ist durch die Unterschiede in der Bedienung der Software und der Hardware anfangs irritiert
HW0607	Kind bevorzugt Bee-Bot statt der Software Focus on Bee-Bot
HW0608	Kind bevorzugt Software Focus on Bee-Bot statt Bee-Bot

Tabelle 14: Kategoriensystem Hardware Usability Test, finale Version

Anmerkung: Unterkategorien, von denen nur eine aus einer bestimmten Gruppe kodiert werden darf (Single Choice), sind an der rechten Seite mit einem Doppelstrich als Gruppe gekennzeichnet.

Die Hinweise zur Kodierung des Kategoriensystems sind in Anhang A4 zu finden.

9.1.6 Ergebnisse

Im Folgenden werden die Ergebnisse der Kodierung des Hardware Usability Tests entlang des Kategoriensystems dargestellt.

9.1.6.1 HW01 Physikalische Eigenschaften des Bee-Bots

Die physikalischen Eigenschaften des Bee-Bots lassen sich nach den Ergebnissen wie folgt darstellen: 7 der 9 beim HWT1 befragten Kinder benannten das Batteriefach des Bee-Bots und 5 der 8 befragten Kinder erkannten, dass eine Stromversorgung von Nöten ist. Nur ein Kind von 19 Kindern (HWT1 und HWT2 zusammen), in diesem Fall ein Mädchen, erwähnte von sich aus, dass der Bee-Bot einen Motor hat oder benötigt. 2 Jungen im Alter von 6 Jahren und ein Mädchen im Alter von 5 Jahren erkannten im Bee-Bot einen Roboter. 12 der 19 Kinder erkannten im Bee-Bot eine Biene und 13 Kinder äußerten zu Beginn des Tests die Vermutung, dass der Bee-Bot fahren kann.

9.1.6.2 HW02 Aktionen des Bee-Bots

Beim Hardware Usability Test thematisierten die Kinder das Piepen des Bee-Bots nicht bzw. sprachen es nicht an. Das Fahren des Bee-Bots wurde wahrgenommen bzw. als eine seiner zentralen Funktionalitäten erkannt, wie auch 13 der 19 Kinder vermuteten. 10 der 19 Kinder thematisierten, erwähnten oder wiesen darauf hin, dass die Augen des Bee-Bots blinken. 7 der 10 Kinder brachten das Augenblinker im Zusammenhang mit einem Tastendruck und 2 der 10 Kinder benannten das Augenblinker als einzige Funktionalität des Bee-Bots. Hier ist anzumerken, dass ein Kind von den beiden auch vermutete, dass der Bee-Bot fahren kann, das jedoch zu einem früheren Zeitpunkt.

9.1.6.3 HW03 Bedienung des Bee-Bots

6 der 19 Kinder betätigten als erste Taste den Einschalter auf der Unterseite des Bee-Bots, 4 der 19 hingegen die GO-Taste am Rücken des Bee-Bots und 3 Kinder die Vorwärts Taste. 12 Kinder schafften es, den Bee-Bot ohne Hilfestellungen und Hinweise seitens der Testleitung einzuschalten, 7 Kinder brauchten dabei Unterstützung, um den richtigen Schiebeschalter zu betätigen.

Um den Bee-Bot zum Fahren zu bringen, ist die Eingabe einer beliebigen Reihenfolge an Richtungsbefehlen über die Richtungstasten gefolgt von einem Tastendruck auf GO von Nöten. Ein Kind versuchte den Bee-Bot zu schieben, ohne dass der Bee-Bot von sich aus losfuhr. Von den 15 Kindern (von insgesamt 19 Kindern), die den Bee-Bot zum Fahren brachten und somit auch eine geeignete Befehls-Reihenfolge eingegeben hatten, betätigten 6 Kinder bewusst sinngemäß die Richtungstasten und 9 Kinder bewusst sinngemäß die GO-Taste. 4 Kinder versuchten während der Fahrt des Bee-Bots eine Richtungsänderung von selbigem herbeizuführen (was nicht vorgesehen ist). 5 Kinder versuchten durch Schieben den fahrenden Bee-Bot entweder zu stoppen oder zu einer Richtungsänderung zu bewegen.

7 Kinder nutzten während des Tests die Pause-Taste, wobei 3 Kinder bei der Verwendung der Pause-Taste irritiert über die Folge / Reaktion des Bee-Bots auf den Tastendruck waren. 5 der 19 Kinder drückten entweder die Clear- oder die Pause-Taste und anschließend die GO-Taste. Dies bewirkte im Falle von Clear und GO einen Stillstand des Bee-Bots, da mit Clear der Speicher gelöscht wurde und mit GO keine Befehle abgearbeitet werden konnten. Die Tastenkombination Pause und GO führte vor allem dann zu Irritation, wenn sonst keine weiteren Befehle eingegeben wurden. Das Verhalten des Bee-Bots bei der Eingabe Pause GO ist die, dass der Bee-Bot zu Beginn der Befehle still steht (pausiert) und anschließend meldet, dass die Abarbeitung der Befehle erfolgt ist (durch Blinken der Augen und, falls der Sound eingeschaltet ist, durch dreimaliges Piepen).

4 Kinder gaben, noch bevor sie den Bee-Bot eingeschalten hatten – meist, weil sie noch nicht wussten, dass der Bee-Bot auf seiner Unterseite zum Einschalten ist – eine geeignete Befehl-Reihenfolge ein, die der Bee-Bot im eingeschalteten Zustand abarbeiten hätte können. 4 der 19 Kinder schafften es nicht, den Bee-Bot zum Fahren zu bringen bzw. eine geeignete Befehls-Reihenfolge einzugeben.

Die Funktion der Richtungstasten am Rücken des Bee-Bots konnten 4 Kinder in eigenen Worten erklären, davon konnten 2 Kinder zusätzlich auch die Funktion der GO-Taste in eigenen Worten erläutern. 3 weitere Kinder konnten die Funktion der GO-Taste erklären, aber nicht die der Richtungstasten.

9.1.6.4 HW04 Herangehensweisen und Eigenschaften des Kindes

Die Mehrheit der Kinder zeigte keine Berührungsängste beim Ausprobieren des Bee-Bots. Von 19 Kindern hatte ein Kind Angst, dass der Bee-Bot, weil er die Form einer Biene hat, stechen kann und ein Kind war so schüchtern, dass hier nicht klar gesagt werden kann, ob Berührungsängste bestanden oder nicht. 15 Kinder zeigten sich experimentierfreudig und 3 Kinder sprachen während der Nutzung des Bee-Bots im Hardware Usability Test mit dem Bee-Bot.

8 Kinder waren während des Tests eher zurückhaltend und verschlossen, wovon 5 Kinder auch kaum bis gar nicht mit der Testleitung interagierten, die anderen 3 Kinder interagierten zwar zurückhaltend aber dennoch mit der Testleitung.

Von den 7 Kindern, bei denen der programmierte Bee-Bot entweder gegen einen Fuß oder eine Hand des Kindes oder den Sitzpolster des Kindes fuhr, versuchten 5 Kinder auszuweichen um die Weiterfahrt des Bee-Bots nicht zu behindern.

9.1.6.5 HW05 Interesse an Bee-Bot

Alle 19 Kinder, die am Hardware Usability Test teilnahmen, waren am Bee-Bot interessiert und die Hälfte der Kinder zeigte starkes Interesse am Bee-Bot.

9.1.6.6 HW06 Analogien zur Software Focus on Bee-Bot 1 bzw. 2

Diese Kategorien kamen nur beim HWT2 (9 Kinder) zum Einsatz. 8 der 9 Kinder erkannten den Bee-Bot in seiner realen Form als Spielfigur der Focus on Bee-Bot 1 bzw. 2 Software wieder. Jedoch nur 3 Kinder erkannten bzw. thematisierten während des Hardware Usability Tests die Analogien in der Bedienung.

6 Kinder konnten die Taste Clear am realen Bee-Bot zeigen bzw. benennen, die und deren Funktionalität bei der Software schon vorgestellt wurden. 4 dieser 6 Kinder konnten die Clear-Taste bewusst sinngemäß verwenden.

3 Kinder erkannten und benannten Unterschiede zwischen dem Hardware Bee-Bot und der Software Focus on Bee-Bot 1 bzw. 2. Ein Kind war zwar durch die Unterschiede sichtlich etwas irritiert bei der Bedienung des realen Bee-Bots, sprach diesen Umstand aber nicht von sich aus an. Ein Kind war durch die Unterschiede irritiert und sprach diese auch an.

Alle Kinder bevorzugten auf die Frage, ob sie lieber mit dem Bee-Bot oder der Software Focus on Bee-Bot 1 bzw. 2 programmieren möchten, den Bee-Bot.

9.1.7 Zusammenfassende Interpretation der Ergebnisse und Schlussfolgerungen

Die physikalischen Eigenschaften des Bee-Bots sind für die Kinder, beim ersten Kennenlernen des Bee-Bots, oft nicht eindeutig und unmittelbar erfassbar.

Die Kinder haben Schwierigkeiten damit, ihre gewonnenen Erkenntnisse zu verbalisieren und einem Erwachsenen zu vermitteln. Sie zeigen lieber direkt an der Hardware, wie sie vorgehen würden.

13 Kinder brachten rein durch Experimentieren den Bee-Bot zum Fahren (HW0307, HW0308, HW0309 und HW0310). 6 Kinder nutzten nicht nur eine geeignete Befehls-Reihenfolge, sondern konnten die Richtungstasten und die GO-Taste bewusst sinngemäß nutzen und zur Bedienung des Bee-Bots einsetzen. 68 % der Kinder, die am Hardware Usability Test teilnahmen, schafften es ohne weitere Unterstützung (außer dem Testsetting, dass sie aufgefordert wurden den Bee-Bot zu nutzen), den Bee-Bot zum Fahren zu bringen und davon konnten 46 % der Kinder den Bee-Bot bewusst sinngemäß (nach kurzen Erläuterungen im Rahmen des Tests) nutzen. Der Bee-Bot ist somit für Kinder im Alter von 3 bis 6 Jahren geeignet, denn Kinder aller in Frage kommenden Altersstufen brachten den Bee-Bot zum Fahren.

Das Einschalten des Bee-Bots stellte die Kinder jedoch vor eine Herausforderung, da nur 6 der 19 Kinder, die am Hardware Usability Test beteiligt waren, dazu zuerst den Schiebeschalter auf der Unterseite des Bee-Bots (ON) nutzten. Viele der Kinder erahnten nach dem ersten Tastendruck (der nicht auf ON erfolgte), dass der Bee-Bot eingeschalten werden muss und drehten ihn instinktiv um. Einige der Kinder drehten den Bee-Bot nach einem kurzen Hinweis um und fanden dort dann den Schiebeschalter. Wenigen Kindern musste die Testleitung genaue Anweisungen zum Einschalten geben.

Wenn Kinder während der Eingabe von Befehlen auch einmal die Pause-Taste nutzten, war ihnen oft nicht klar, wozu diese vorhanden ist (HW0312 und HW0313), denn der Pause-Befehl wird vom Bee-Bot wie ein Richtungsbefehl abgearbeitet, so dass ein „Warten“-Befehl einprogrammiert werden kann. Durch die nicht unmittelbare Reaktion des Bee-Bots auf die Pause-Taste und ihr gleiches Aussehen wie das der Clear-Taste, waren die Kinder bei der Ausführung der Befehle des Bee-Bots stets etwas irritiert, wenn dieser einen Ausführungszyklus lang stehen blieb und „wartete“. Die Spezialtasten Clear und Pause sorgten bei den Kindern zumindest für Irritationen und gerade die Clear-Taste war für die Kinder in ihrer Funktionalität nicht intuitiv und leicht zu verstehen.

Mit dem Hardware Usability Test zeigte sich, dass die Kinder kaum Berührungsängste bezüglich des Bee-Bots aufweisen. Einzelfälle sind zwar in der Praxis zu berücksichtigen und durch entsprechende Interaktionen zu vermeiden, daraus kann aber nicht verallgemeinert werden, dass der Bee-Bot für 3- bis 6-jährige Kinder, egal welchen Geschlechts, einschüchternd oder verängstigend wirkt.

79 % der teilnehmenden Kinder zeigten sich durchaus experimentierfreudig (HW0404) und erkundeten den Bee-Bot zum großen Teil selbständig oder mit kurzen Aufforderungen. 42 % der Kinder waren während des Hardware Usability Tests eher zurückhaltend und verschlossen (HW0405). Das kann zum einen am Setting, das für einige Kinder ungewohnt

gewesen sein dürfte, und zum anderen auch in der Person der einzelnen Kinder liegen. 32 % der involvierten Kinder interagierten kaum mit der Testleitung und auch mehrmaliges Nachfragen und motivierende Worte brachten in dieser Hinsicht kaum eine Veränderung (HW0407). 2 Kinder waren anfangs eher zurückhaltend, nachdem sie aber Vertrauen in die Situation gefasst hatten, brachten sie sich stärker in die Interaktion ein und beteiligten sich aktiver am Testgeschehen. Die Wahl der Methode des halbstandardisierten Cognitive Walkthrough erwies sich als richtig, denn so konnte sichergestellt werden, dass annähernd alle Kinder zu den gewünschten Themen Stellung nehmen konnten. Zum anderen stellte sich jedoch, wie zu erwarten war (siehe Kapitel 5.3.1), die Thinking Aloud und Talk Aloud Komponenten des Cognitive Walkthrough als eine Herausforderung im Testsetting dar – die Kinder mussten durch sehr viele Nachfragen und Aufmunterungen zu Aussagen motiviert werden.

3 Kinder begannen mit dem Bee-Bot zu sprechen und versuchten so, den Bee-Bot zu steuern bzw. zu programmieren.

Alle beteiligten Kinder waren am Bee-Bot interessiert und die Hälfte war sogar sehr interessiert am Bee-Bot. Somit erscheint der Bee-Bot als adäquater Bodenroboter zum Erlernen von Programmieren für 3- bis 6-jährige Kinder, da dieser das Interesse bei den Kindern weckt.

Fast alle Kinder, die zuerst die Software kennengelernt hatten und in weiterer Folge den realen Bee-Bot ausprobieren konnten, erkannten den realen Bee-Bot als Spielfigur aus der Software wieder. Das ist nachvollziehbar, da der Bee-Bot in seiner virtuellen Ausprägung sehr dominant auf der Software-Oberfläche zur Verfügung steht.

Dass nur 3 der 9 Kinder, also ein Drittel, Analogien in der Bedienung des Bee-Bot erkannten, verwundert hingegen, da bis auf minimale Unterschiede in der Optik, der Bee-Bot ähnlich zu steuern und zu programmieren wäre, siehe dazu auch Kapitel 9.1.6.6. Wahrscheinlich waren die Kinder davon irritiert, dass der reale Bee-Bot seine Bedienelemente am Rücken hat und nicht in einem separaten Navigationsbereich, wie in der Software.

Die Clear-Taste stellte die Kinder immer wieder vor Herausforderungen, wie auch in der Kategorie „HW03 Bedienung des Bee-Bots“ festgestellt wurde. Hier konnten zwar 6 der 9 Kinder die Taste benennen, doch nur 4 konnten sie bewusst sinngemäß anwenden, wobei davon ein Kind unmittelbar nach Clear die GO-Taste drückte, was sowohl bei der Hardware als auch bei der Software zum Löschen des Befehlspeichers und keiner Abarbeitung von Befehlen führt, da der Speicher leer ist.

3 Kindern fielen Unterschiede in der Bedienung des Bee-Bots zwischen der Hardware und der Software auf und sie benannten diese auch, wobei nur ein Kind mit den Unterschieden anfangs nicht zurechtkam. Ein weiteres Kind erschien irritiert durch die Unterschiede in der Bedienung, thematisierte diese jedoch nicht, so dass es eine Vermutung der Testleitung ist, dass dieses Kind durch die Unterschiede in der Bedienung irritiert war.

Alle 9 Kinder, die am HWT2 teilnahmen, wollten mit dem Hardware Bee-Bot programmieren und nicht mit der Software. Den meisten davon gefiel der Hardware Bee-Bot optisch besser, manche fanden das Drücken darauf lustiger und andere wiederum wollten den Bee-Bot einfach nur angreifen können.

9.1.7.1 Exemplarische Unterschiede zwischen HWT1 und HWT2

Wie zu erwarten war, war für 89 % der Kinder, die zuerst die Software Focus on Bee-Bot 1 bzw. 2 (HWT2) kennengelernt hatten klar, dass der Bee-Bot fahren kann (HW0106), was jedoch nur 50 % der Kinder, die noch nie mit dem Bee-Bot zu tun hatten (HWT1), eingangs vermuteten.

Vor allem jene Kinder, welche den Bee-Bot zum ersten Mal in den Händen hielten (HWT1) thematisieren viel eher das Augenblinker des Bee-Bots, 70 %, als diejenigen, die zuerst die Software kennenlernten, 33 % (HW0201). Da der Bee-Bot in der Software nicht mit den

Augen blinkt, scheint das den Kindern, welche bereits die Software kannten, nicht aufgefallen oder nicht wichtig erschienen zu sein.

100 % der Kinder aus der Gruppe HWT2 brachten den Bee-Bot zum Fahren, wohingegen nur 60 % der Kinder aus der Gruppe HWT1 den Bee-Bot tatsächlich zum Fahren brachten (HW0307). Das liegt daran, dass der Bee-Bot ähnlich wie in der Software gesteuert wird und die Kinder aus der Gruppe HWT2 bereits auf diese (Lern-)Erfahrung zurückgreifen konnten. Da 60 % der Kinder, die den Bee-Bot noch nie gesehen hatten, ihn ebenso zum Fahren brachten, erscheint die Hardware für die Zielgruppe der 3- bis 6-jährigen Kinder geeignet.

Knapp 60 % der Kinder, die den Bee-Bot mit der entsprechenden Software kennengelernt hatten (HWT2), konnten die GO-Taste bewusst sinngemäß nutzen, aber auch 40 % der Kinder, die den Bee-Bot noch nie gesehen hatten (HWT1), konnten die GO-Taste bewusst sinngemäß nutzen (HW0309). Bei der bewussten sinngemäßen Nutzung der Richtungstasten war ähnliches zu beobachten: 44 % aus dem HWT2 konnten diese entsprechend nutzen, 20 % aus dem HWT 1 konnten die Richtungstasten bewusst sinngemäß nutzen (HW0310). Das ist vor allem dem Lerneffekt der Nutzung der Software Focus on Bee-Bot 1 bzw. 2 geschuldet, der bei der Gruppe HWT2 auftreten sollte und auftrat. Interessanter Weise sprachen 3 Kinder aus der Gruppe, die zuerst die Software kennengelernt hatten, mit dem Bee-Bot, wohingegen kein Kind aus der anderen Gruppe mit dem Bee-Bot sprach (HW0403). Auch schoben 44 % der Kinder aus der Gruppe HWT2 den Bee-Bot, dem gegenüber kamen nur 20 % der Kinder aus dem HWT1 auf diese Idee.

Etwas überraschend ist auch die Tatsache, dass ein Drittel der Kinder, die den Bee-Bot von der Software her kannten, während der Abarbeitung der Befehle durch den Bee-Bot versuchte selbigen durch weiteres Drücken von Richtungstasten in seiner Befehlsabarbeitung zu beeinflussen. Dies funktioniert weder in der Software noch mit der Hardware. Hingegen versuchte nur ein einziges Kind, das den Bee-Bot zum ersten Mal sah, ihn durch weiteres Bedienen von Richtungstasten in der Befehlsabarbeitung zu beeinflussen. Hier kann nur vermutet werden, warum das so ist. Da in der Software ein weiterer Tastendruck während der Abarbeitung der Befehle keine Auswirkungen hat und auch nichts dabei „kaputt“ gehen kann, könnten die Kinder in Versuchung geraten sein bei der Hardware ebenso auszuprobieren, welche Folgen ein Betätigen einer Richtungstaste während der Abarbeitung der Befehle mit sich bringt. Ein anderer Grund könnte sein, dass die Kinder der Gruppe HWT2 bereits vertrauter mit dem Bee-Bot waren und sich daher eher experimentierfreudig zeigten, was sich auch in Kategorie HW0404 ablesen lässt. Hier zeigt sich, dass die Kinder aus der Gruppe HWT2 leicht experimentierfreudiger waren (89 %), als die Kinder der Gruppe HWT1 (70 %).

Das Interesse am Bee-Bot ist in beiden Gruppen gleich stark ausgeprägt – hier sind keine Unterschiede wahrzunehmen.

Zusammengefasst kann festgehalten werden, dass den Kindern der Gruppe HWT2 die sinngemäße Bedienung des Bee-Bots etwas leichter fiel, als den Kindern der Gruppe HWT1. Der Transfer der gewonnenen Erfahrungen und des Erlernen mit der Software auf die Hardware klappte bei den meisten Kindern soweit, dass sich Vorteile bzw. positive Unterschiede bezüglich der erstmaligen Benutzung und Bedienung der Hardware gegenüber den Kindern aus der Gruppe HWT1, die den Bee-Bot noch nie gesehen hatten, zeigten.

9.1.7.2 Exemplarische geschlechterspezifische Unterschiede

Betrachtet man die Ergebnisse unter dem Aspekt, ob es Unterschiede zwischen den Geschlechtern gibt, dann muss man sehr vorsichtig sein mit der Interpretation der Daten. Wie in Kapitel 2.5.1 dargelegt, zeigen sich schon im Kindergarten bei den 3- bis 6-Jährigen so manche Unterschiede bei der Herangehensweise, dem Verhalten oder der Nutzung von Technik.

90 % der in den Hardware Usability Test involvierten Jungen benötigten keine Hilfe zum Einschalten des Bee-Bots, wohingegen zwei Drittel der Mädchen Hilfestellungen oder

Hinweise benötigten. Jungen tun sich etwas leichter bei der Erfassung der Bedeutung von Tasten, da sie diese eher bewusst sinngemäß nutzen (HW0309 und HW0310) als Mädchen. Wohingegen gleich viel Mädchen wie Jungen den Bee-Bot zum Fahren brachten (HW0307), ihn schieben wollten (HW0304) oder einen Zusammenhang zwischen Tastendruck und dem Blinken der Augen herstellen konnten (HW0202).

Mädchen sind bei diesem Hardware Usability Test zurückhaltender, was das Ausprobieren des Bee-Bots, die Herangehensweise und die Bedienung angeht. Bei der generellen Zurückhaltung im Testsetting ist das Verhältnis Mädchen zu Jungen ausgeglichen (HW0405). Jedoch interagierten Mädchen eher mit der Testleitung als Jungen (HW0407).

Jungen äußerten eher hohes Interesse am Bee-Bot, wohingegen die Mädchen eher nur Interesse bekundet haben (HW05) – das könnte jedoch auf das eher zurückhaltende Verhalten der Mädchen während des Hardware Usability Tests zurückgeführt werden. Dieses Ergebnis ist daher mit besonderer Vorsicht zu betrachten, da die Kinder nicht explizit nach ihrem Interesse gefragt wurden, sondern aus der Beobachtung des Tests dieses Kriterium kodiert wurde.

In der Kategorie HW06 konnten keine geschlechterspezifischen Unterschiede wahrgenommen werden.

Keine generellen Aussagen können über folgende Aspekte getroffen werden, interessant im Kontext der durchgeführten Forschungsarbeit sind diese jedoch schon:

- Nur Jungen redeten mit dem Bee-Bot (HW0403).
- Als einziges Kind erwähnte ein Mädchen, dass der Bee-Bot einen Motor hat (HW0103).
- Als einziges Kind erwähnte ein Mädchen, aber ein anderes Mädchen als beim vorhergegangenen Punkt, dass es Angst vor dem Bee-Bot hat, da dieser stechen kann, weil er eine Biene sei (HW0402).
- Alle Mädchen, die danach gefragt oder darauf im Verlauf des Gesprächs geführt wurden, erkannten, dass der Bee-Bot eine Stromversorgung benötigt. Bei den Jungen nannte nur einer von den 4 dazu befragten Jungen, dass der Bee-Bot eine Stromversorgung benötigt. Auch dieses Ergebnis ist mit besonderer Vorsicht zu genießen, da nicht alle teilnehmenden Kinder an dem Hardware Usability Test auf Grund der gewählten Testmethode danach gefragt wurden.

9.1.7.3 Beitrag der Hardware Usability Evaluierung zur frühen Technikbildung in Informatik

Die Kinder trainieren durch die Teilnahme an den Lehr-Lern-Einheiten unterschiedliche Strategien zur Bewältigung von Problemstellungen (korrespondierend zu den Lernergebnissen aus Kapitel 8.2) in der Informatik und Technik allgemein. Dazu zählen durch „trial and error“ ein ihnen unbekanntes Objekt zu erkunden (vgl. LE2.4), einen Zusammenhang zwischen den Aktionen, die sie mittels Tastendrucks auslösen und den Reaktionen, die der Bodenroboter zeigt, herzustellen (vgl. LE2.4) sowie an themenrelevantes Vorwissen anzuknüpfen (vgl. LE1.7).

Im Zuge der Evaluierung der Hardware Usability üben die Kinder genau zu beobachten und die Beobachtungen in Worte zu fassen, offene Frage- und Problemstellungen zu beantworten bzw. zu bearbeiten (vgl. LE3.4) und ihr Handeln zu kommentieren (vgl. LE3.5). Ebenso beschäftigen sie sich damit Parallelen zwischen der korrespondierenden Hard- und Software zu erkennen und zu benennen.

Durch die Testleitung werden die Kinder animiert, informatische und technische Begriffe in eigenen Worten wiederzugeben (vgl. LE2.2 und LE1.1) und mit der Testleitung selbst zu interagieren. Weiter wird der Grundstein gelegt, dass die teilnehmenden Kinder die grundlegende Funktionalität eines Bodenroboters erklären (vgl. LE2.5) und ihn zur Lösung von Problemstellungen bedienen (vgl. LE2.6) können.

9.2 Evaluierung Software Usability

Der Software Usability Test stellt den zweiten Teil der Usability Evaluierung des Bee-Bots und der Software Focus on Bee-Bot 1 bzw. Focus on Bee-Bot 2 dar.

Beim Softwaretest 1 (SWT1) nahmen neun Kinder aus dem Kindergarten D an der Evaluierung der Usability der Software Focus on Bee-Bot 1 bzw. 2 teil. Diese Kinder hatten zuvor, nach Angaben der Kinder und der betreuenden Pädagoginnen und Pädagogen, noch nie Kontakt mit der Software, jedoch mit dem Bee-Bot. Sie verfügen daher bereits über erste Kenntnisse vom realen Bee-Bot. Im Rahmen des Software Usability Tests ist folglich auch ein möglicher Transfer des Wissens der Kinder, welches sie im Rahmen der Hardware Nutzung erworben haben, auf die Bedienung der Software von Interesse.

Beim Softwaretest 2 (SWT2) nahmen neun Kinder des Kindergartens E an der Evaluierung teil. Diese Kinder hatten ebenfalls, nach Angaben der Kinder und der betreuenden Pädagoginnen und Pädagogen, noch nie Kontakt mit der Software Focus on Bee-Bot 1 bzw. Focus on Bee-Bot 2, aber auch noch keinen Kontakt mit dem realen Hardware Bee-Bot.

Zu Beginn stellt die Autorin ihre eigene Usability Evaluierung der Software Focus on Bee-Bot 2 in Form eines Experten-Reviews, basierend auf den Guidelines von Pohl & Schmalzl (2010) vor (Kapitel 9.2.1). In Kapitel 9.2.2 beschreibt die Verfasserin der gegenständlichen Arbeit das Testsetting und die Vorgehensweise bei der Evaluierung der Software Usability mit den Kindergartenkindern im Kindergarten. Kapitel 9.2.3 behandelt die Fragestellung der Software Evaluierung und in Kapitel 9.2.4 ist der Leitfaden zur zusammenfassenden Transkription zu finden. Folgend in Kapitel 9.2.5 legt die Autorin die Entwicklung der Kategorien dar. In Kapitel 9.2.6 präsentiert die Verfasserin der gegenständlichen Arbeit die Ergebnisse der Software Usability Evaluation, die sie in Kapitel 9.2.7 konsolidiert und interpretiert.

9.2.1 Expertenreview Software Focus on Bee-Bot 2

Die Verfasserin der gegenständlichen Arbeit besuchte die Lehrveranstaltung „Computerunterstütztes Lernen“ und fertigte im Rahmen dieser ein Review-Dokument zur Usability-Evaluierung der Software Focus on Bee-Bot 2 an. Dazu orientierte sie sich an den von Pohl & Schmalzl (2010) entwickelten Guidelines zur Evaluierung von Lernprogrammen.

Im Zuge der Evaluation betrachtete die Autorin mehrere Dimensionen. Dazu zählen

- Modularisierung,
- Interaktivität,
- Multimedia-Elemente und Visualisierung,
- Mobiles Lernen,
- Kooperatives Lernen,
- Lernplattformen sowie
- Genderspezifische Guidelines und Diversity.

Im Folgenden werden die wesentlichen Erkenntnisse zusammengefasst dargestellt. Zu Beginn ist anzumerken, dass die Software, konzipiert für Vor- und Volksschulkinder, für die grundlegende Bedienung zwar keine, aber für die Nutzung des vollen Funktionsumfangs Schreib- und Lesefähigkeiten der Kinder erfordert. Somit ist in Folge die Software nur bedingt für Kindergartenkinder und frühe Volksschulkinder zur selbständigen Nutzung geeignet.

Unter dem Aspekt Modularisierung fällt auf, dass zwar die verwendeten Icons mit Tool-Tipps versehen sind, jedoch oftmals von den Kindern, die die Software nutzen, noch nicht gelesen werden können. Auch die abgebildeten Wissensstrukturen sind über weite Strecken generalisiert - die Navigation zwischen den Aufgaben ist sehr verallgemeinert: vorhergehende Aufgabe, nächste Aufgabe.

Die Software Focus on Bee-Bot 2 stellt keine Möglichkeit zur Rückverfolgung bereits erledigter Aufgaben zur Verfügung. So kann auch nie der eigene Lernstand mit Hilfe der Software bestimmt werden. Auch sind die Aufgaben in keiner (hierarchischen) Struktur angeordnet und Zusammenhänge zwischen den Aufgaben bzw. thematische Schwerpunkte sind nicht unmittelbar ersichtlich.

Der Grad der Interaktivität der Software könnte höher sein. Wenn eine Aufgabe von einer Nutzerin oder einem Nutzer richtig gelöst wurde, erhält die- oder derjenige eine entsprechende Rückmeldung. Bei nicht richtig gelösten Aufgaben oder Fehlern erhält die Nutzerin bzw. der Nutzer keine Rückmeldung. Es liegen auch keine Tutorials, Anleitungen oder Lösungen zu den Aufgaben in der Software vor. Die Aufgabenstellungen sollten idealerweise für Kinder, die (noch) nicht lesen können, vorgelesen werden, was in der Software Focus on Bee-Bot 2 nicht der Fall ist. Die Software animiert die Nutzerinnen und Nutzer zum eigenständigen „Fragen formulieren“ und bietet zahlreiche Möglichkeiten zum selbständigen Lernen mit der Software.

Als Analogie bildet die Software den realen Hardware Bee-Bot genau gleich in seiner Funktionsweise und Optik in der Software ab. Irritierend ist jedoch, dass bei Programmstart der Bee-Bot 180° gedreht im Vergleich zum Navigationsbereich (wo der virtuelle Bee-Bot programmiert wird) steht. Auch die Tatsache, dass der virtuelle Bee-Bot auf seinem Rücken die Tasten zur Bedienung abgebildet hat, diese in der Software aber nicht bedienbar sind, führt zu anfänglicher Irritation. Grundsätzlich ist die Software auf das Wesentliche, die Simulation der Realität, beschränkt. Die Benutzerschnittstelle ist überwiegend mit grafischen Elementen gestaltet, die größtenteils Text überflüssig machen, nur Fehlermeldungen und die Aufgabenstellung sind in Text ausgestaltet. Die dargestellten Spielmatte können, da es sich um 3D Objekte handelt, im Raum gedreht und somit der Blickwinkel auf den virtuellen Bee-Bot verändert werden kann.

Die Software Focus on Bee-Bot ist nicht explizit für kooperatives Lernen gestaltet, jedoch kann sie durchaus von kleinen Teams, mit zwei oder drei Kindern, genutzt werden, z.B. in dem sich ein Kind eine Aufgabenstellung überlegt, die von dem zweiten Kind im Team mit Hilfe der Software gelöst werden soll.

9.2.2 Beschreibung des Tests

9.2.2.1 Testziel

Mit dem Software Usability Test der Software Focus on Bee-Bot 1 bzw. 2 gilt es nicht nur Usability-Schwächen der Software zu identifizieren, sondern auch herauszufinden, ob 3- bis 6-jährige Kinder in der Lage sind, diese aufzuzeigen und damit an weiterführenden Aktivitäten im Child-Centered-Design mitwirken können. Darüber hinaus soll die Evaluierung der Software Usability einen Beitrag zur frühen Technikbildung in Informatik (siehe Kapitel 9.2.7.3) leisten.

9.2.2.2 Kurzbeschreibung des Settings

Die Autorin führte die Software-Usability-Tests SWT1 und SWT2 in den Kindergärten D und E unter Einsatz des folgenden Settings durch: Die Tests fanden in einer den Kindern vertrauten Umgebung statt, in einem ruhigen Raum, wo ungestört mit der jeweiligen Software gearbeitet werden konnte. Ein Kind (K) nach dem anderen nahm vor einem Notebook Platz. Dem Kind stand für die Bedienung des Notebooks eine Maus zur Verfügung. Neben dem Kind (K) saß die Testleitung (TL). Mittels eines separaten Tonaufnahmegeräts und einer Videokamera erfolgte eine vollständige Aufzeichnung des Software Usability Tests. Die Autorin zeichnete mit der Software Morae® von TechSmith den Bildschirm-Inhalt auf, um die Mausbewegungen und Klick-Aktionen der Kinder dokumentieren zu können. Die wichtigste Datenquelle dieses Software-Usability Tests ist die Aufzeichnung des Bildschirm-Inhalts, das Screenrecording, wohingegen die Ton- und

Videoaufzeichnungen als Backup dienen, wenn Inhalte akustisch nicht verständlich sein sollten oder es zu einem Ausfall eines der drei Aufnahmesysteme kommen sollte.

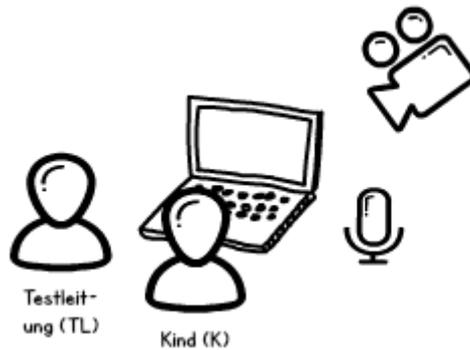


Abbildung 27: Testsetting Software Usability Test

9.2.2.3 Demografische Daten

Die Testgruppen des SWT1 und SWT2 setzten sich wie folgt aus Kindergartenkindern zusammen:

	Weiblich	Männlich
SWT1	4	5
3 Jahre	0	0
4 Jahre	0	2
5 Jahre	1	1
6 Jahre	3 ³⁵	2
SWT2	4	5
3 Jahre	0	1
4 Jahre	1	1
5 Jahre	3	0
6 Jahre	0	3

Tabelle 15: Verteilung Alter und Geschlecht der Probandinnen und Probanden

Die Testgruppe SWT1 aus Kindergarten D bestand somit aus vier Mädchen und fünf Jungen im Alter von 4 bis 6 Jahren. Die Testgruppe SWT2 aus Kindergarten E bestand aus vier Mädchen und fünf Jungen im Alter von 3 bis 6 Jahren.

9.2.2.4 Eingesetzte Methoden und Vorgehensweise

Die Verfasserin der gegenständlichen Arbeit setzte für den Software Usability Test einen halbstandardisierten Cognitive Walkthrough ein. Einerseits sollten die Kinder durch freies Experimentieren und Ausprobieren die jeweilige Software kennenlernen und erkunden können. Andererseits musste sichergestellt werden, dass alle relevanten Themen behandelt werden. Da Kinder sehr individuell auf die jeweilige Situation reagieren, muss die Testleitung Fingerspitzengefühl beweisen und flexibel in Bezug auf die Fragestellung vorgehen. Sollte ein Kind eine Frage nicht beantworten können, muss die Frage von der Testleitung beantwortet werden. Die Kinder würden bei allfälligen Misserfolgen zu schnell die Lust und Motivation verlieren. Weiter darf dem Kind nicht das Gefühl vermittelt werden, es wäre nicht in der Lage die Aufgabenstellung korrekt zu bearbeiten.

³⁵ Ein Mädchen war zum Zeitpunkt des Tests noch nicht 6 Jahre alt, da sie jedoch ebenso kurz vor der Einschulung stand, wie auch die anderen 6-jährigen Kinder, wurde sie der Gruppe der Schulanfängerinnen und -anfänger im Alter von 6 Jahren zugeordnet.

Die Aufnahmen der Bildschirminhalte dienten als Basis für die weitere Auswertung. Die Verfasserin der gegenständlichen Arbeit kombinierte für eine qualitative Auswertung, wie beim Hardware Usability Test (siehe Kapitel 9.1.3), die Videographie (siehe 7.1.1.4) mit der qualitativen Inhaltsanalyse nach Mayring (siehe Kapitel 7.2). Die Autorin machte das Videomaterial durch eine zusammenfassende Transkription (siehe Kapitel 9.2.4) lesbar und bereitete diese in weiterer Folge durch Kodierung (siehe 9.2.5) auf. Zur Objektivierung der Kodierung (siehe Kapitel 7.2.2) entwickelte die Autorin einen Leitfaden zur zusammenfassenden Transkription und ein Kategoriensystem zur Kodierung (siehe Kapitel 9.2.4 und 9.2.5). Sollte ein Aspekt im Transkript nicht auf Anhieb eindeutig einer Kategorie zuordenbar sein, konnte die Verfasserin der gegenständlichen Arbeit die gesamte Szene nochmals im Videomaterial betrachten um zu einer Entscheidung zu kommen. Die Ermittlung und Interpretation der Ergebnisse aus der Kodierung bildeten den Abschluss dieses Tests (siehe Kapitel 9.2.6 und 9.2.7).

Die Software Usability Tests fanden pro Kindergarten jeweils an einem Tag für alle Probandinnen und Probanden hintereinander statt. Ein Test dauerte zwischen 15:49 Minuten und 32:33 Minuten. Um die Teilnahme aller Kinder einer Gruppe (SWT1 und SWT2) an einem halben Tag an dem Software Usability Test sicherzustellen, fanden immer zwei Tests parallel statt. Bei den Software Usability Tests jeweils einer Hälfte der Kinder pro Kindergarten übernahm die Verfasserin der gegenständlichen Arbeit die Testleitung. Die Begleitung beim Software Usability Test der anderen Hälfte der Kinder erfolgte durch einen weiteren Tester, der die Tests nach Vorgaben der Autorin durchführte. Die Testleitung verzichtete jeweils in Anwesenheit der Kinder weitestgehend auf das Wort „Test“, um unter allen Umständen zu vermeiden, dass für die Kinder eine Art Prüfungssituation entstehen könnte.

9.2.3 Fragestellung

Die Software Usability Evaluierung umfasste folgende halbstandardisierte Fragestellung, deren Fragen als Leitfragen zu verstehen sind. Es bestand die Möglichkeit für die Testleitung weitere Fragen, die im Kontext geeignet erscheinen, zu stellen und auch die eine oder andere Frage situativ begründet wegzulassen.

Die Autorin passte die jeweilige Fragestellung an die beiden Gruppen an, da die Gruppe SWT1 bereits den Hardware Bee-Bot kennengelernt hatte und die Gruppe SWT2 den Hardware Bee-Bot noch nicht kannte (siehe auch Kapitel 9.1.1).

9.2.3.1 Fragestellung Gruppe SWT1

I) **Fragen / Aufgaben Focus on Bee-Bot Software 1**

- 1) Bitte kannst du mir beschreiben was Du siehst.
- 2) Siehst du einen Zusammenhang zwischen dem, was wir im Kurs mit euch bis jetzt gemacht haben und dem, was du hier siehst? (hier kann man den Hinweis geben, dass es sich um Bee-Bot handelt)
- 3) Wie kann der Bienenroboter gesteuert werden?
- 4) Gibt es hier bei den Buttons (Tasten) einen Unterschied zu den Tasten beim echten Bienenroboter?
- 5) Hast du diese Matte schon einmal gesehen?
- 6) Bitte kannst du auf der Matte, auf der die Biene steht mit der Biene einen Schritt vorwärts gehen.
- 7) Wie kann die Biene hier im Programm „vergessen“? Bitte drücke vergessen und stelle die Biene wieder an den Start.
- 8) Glaubst du kann die Biene von dort aus, wo sie jetzt steht, zwei Schritte vorwärts fahren?
- 9) Glaubst du kann die Biene von der Matte runterfahren?

- 10) Der Bienenroboter soll drei Schritte nach rechts gehen.
- 11) Findest du ein Symbol, mit dem man sich die Einzelschritte der Biene anzeigen lassen kann?

II) Fragen / Aufgaben Focus on Bee-Bot Software 2

- 1) Welche Unterschiede zum Programm vorher fallen dir auf?
- 2) Wie kann der Bienenroboter gesteuert werden?
- 3) Gibt es hier bei den Buttons (Tasten) einen Unterschied zu den Tasten beim echten Bienenroboter?
- 4) Bitte kannst du auf der Matte, auf der die Biene steht mit der Biene einen Schritt vorwärts gehen.
- 5) Wie kann die Biene hier im Programm „vergessen“? Bitte drücke „vergessen“ und stelle die Biene wieder an den Start.
- 6) Glaubst du kann die Biene von dort aus, wo sie jetzt steht zwei Schritte vorwärts fahren?
- 7) Glaubst du kann die Biene von der Matte runterfahren?
- 8) Der Bienenroboter soll drei Schritte nach links gehen.
- 9) Findest du ein Symbol, mit dem man sich die Einzelschritte der Biene anzeigen lassen kann?
- 10) Falls in beiden Fällen (Software 1 und Software 2) keine Schritte gemacht werden konnten kurze Erklärung der Software

III) Lieblingssoftware und Aufgabenstellung mit Matte

- 1) Welches von den beiden Programmen gefällt dir besser? Warum?
- 2) Im Lieblingsprogramm: Such dir eine Matte aus mit der du arbeiten möchtest.
- 3) Schwierige Route mit zumindest zwei Kurven

IV) Zeigen des Constructa-Bots

9.2.3.2 Fragestellung Gruppe SWT2

Die Kinder im Kindergarten E kannten den realen Bee-Bot vor der Durchführung des SWT2 im Unterschied zu den Kindern des Kindergartens D vor der Durchführung des SWT1 (siehe auch Kapitel 9.1) nicht. Aus diesem Grund stellte die Testleitung Fragen, die sich auf die Vorerfahrung der Kinder mit dem realen Bee-Bot beziehen, im Kindergarten E nicht. Konkret waren dies die im Kindergarten D beim SWT1 gestellten Fragen mit den Nummern I2), I4), I5) und II3). Die Fragestellung im Kindergarten E beim SWT2 ergänzte die Autorin anstelle dessen um Fragen, die die allgemeine Vorerfahrung der Kinder mit Software, mit Robotern bzw. Erwartungen der Kinder an die Software Focus on Bee-Bot erfassen. Es sind dies die im Kindergarten E beim SWT2 gestellten Fragen mit den Nummern I2a), I2b) und I2c. Wo es für das Testsetting sinnvoll erschien, stellte die Testleitung Fragen des SWT1 im SWT2 an anderer Stelle, es war dies die Frage mit der Nummer I10) des SWT1, die im SWT2 im ersten Teil als siebte Frage gestellt wurde. Die Frage I7) des SWT1 formulierte die Verfasserin der gegenständlichen Arbeit beim SWT2 geringfügig um und stellte sie an Position I6) des Testsettings im SWT2.

I) Fragen / Aufgaben Focus on Bee-Bot Software 1

- 1) Bitte kannst du mir beschreiben was du siehst.
- 2) Frage besteht aus Unterfragen:
 - a) Was glaubst du kann die Biene machen?
 - b) Hast du so ein ähnliches Programm schon einmal gesehen?
 - c) Glaubst du ist das ein Roboter?

- 3) Wie kann der Bienenroboter gesteuert werden?
- 4) Bitte kannst du auf der Matte, auf der die Biene steht mit der Biene einen Schritt vorwärts gehen.
- 5) Der Bienenroboter soll drei Schritte nach rechts gehen. (Frage an anderer Stelle als in Kindergarten D – damit sollen die Kinder zur Erkenntnis angeregt werden, dass die Biene die vorhergehende Befehlsspeicherung noch gespeichert hat)
- 6) Wie kann die Biene Dinge, die du ihr speicherst „vergessen“? Bitte drücke „vergessen“ und stelle die Biene wieder an den Start.
- 7) Glaubst du kann die Biene von dort aus, wo sie jetzt steht, zwei Schritte vorwärts fahren?
- 8) Glaubst du kann die Biene von der Matte runterfahren?
- 9) Findest du ein Symbol, mit dem man sich die Einzelschritte der Biene anzeigen lassen kann?

II) Fragen / Aufgaben Focus on Bee-Bot Software 2

- 1) Welche Unterschiede zum Programm vorher fallen dir auf?
- 2) Wie kann der Bienenroboter gesteuert werden?
- 3) Bitte kannst du auf der Matte, auf der die Biene steht mit der Biene einen Schritt vorwärts gehen.
- 4) Wie kann die Biene hier im Programm „vergessen“? Bitte drücke „vergessen“ und stelle die Biene wieder an den Start.
- 5) Glaubst du kann die Biene von dort aus, wo sie jetzt steht zwei Schritte vorwärts fahren?
- 6) Glaubst du kann die Biene von der Matte runterfahren?
- 7) Der Bienenroboter soll drei Schritte nach links gehen.
- 8) Findest du ein Symbol, mit dem man sich die Einzelschritte der Biene anzeigen lassen kann?
- 9) Falls in beiden Fällen (Software 1 und Software 2) keine Schritte gemacht werden konnten kurze Erklärung der Software

III) Lieblingssoftware und Aufgabenstellung mit Matte

- 1) Welches von den beiden Programmen gefällt dir besser? Warum?
- 2) Im Lieblingsprogramm: Such dir eine Matte aus mit der du arbeiten möchtest.
- 3) Schwierige Route mit zumindest zwei Kurven

IV) Zeigen des Constructa-Bots

9.2.4 Leitfaden zur zusammenfassenden Transkription

Die zusammenfassende Transkription des Videomaterials der Software Usability Tests erfolgt analog zu der zusammenfassenden Transkription der Hardware Usability Tests. Für jeden Test erstellt die Verfasserin der gegenständlichen Arbeit eine Transkriptions-Datei, die zu Beginn das Datum des Tests, das Pseudonym des Kindes, den Dateinamen des Videos und den anonymisierten Ort des Tests enthält.

Durch die Transkription beschränkt die Autorin den Inhalt des Ausgangsmaterials auf das Wesentliche (siehe Kapitel 7.2.3.1), eliminiert Irrelevantes oder ersetzt dieses durch Verallgemeinerungen wie „Unterbrechung“. Die Verfasserin der gegenständlichen Arbeit klassifiziert all das als wesentlich, was die Fragestellung, Aktionen vom jeweiligen Kind und von der Testleitung sowie die Interaktionen zwischen ihnen und der Software Focus on Bee-Bot 1 bzw. 2 betrifft. Alles, bei dem die Software Focus on Bee-Bot 1 bzw. 2 im Zentrum des Geschehens steht, erfasst die Autorin so schriftlich und zusätzlich zum Videomaterial.

Die Verfasserin der gegenständlichen Arbeit transkribiert Aussagen vom jeweiligen Kind und der Testleitung wortwörtlich, erfasst die Aktionen und Beobachtungen der Szenen durch Paraphrasierungen und erstellt zur Dokumentation pro Test folgende Tabelle (siehe Abbildung 28).

Minute	Frage Nr.	Aussage Kind	Aussage TL	Beobachtung	Aktion Kind	Aktion TL
16:45			Mhm (Ja)			
		Und jetzt da			K navigiert zum Linksdreh-Button im Navigationsbereich.	
			Wohin fährst du dann?		K klickt mit der linken Maustaste.	
				Gemäß der Funktionalität von Einzelschritte anzeigen fährt die Biene unmittelbar den gespeicherten Befehl, eine Vierteldrehung nach Links. Sie steht dann am fünften Feld von links in der zweiten Reihe von oben mit Blick nach oben.		
		Ah	Nach links.			
16:51						K klickt nochmals mit der linken Maustaste.

Abbildung 28: Dokumentationsschema zur Transkription

Die Spalte Minute dient dem Vermerk der Zeitangabe der betreffenden Stelle im Video, um ein einfaches Auffinden derselben im Ausgangsmaterial zu ermöglichen. Die Zeitangabe erfolgt immer bei wesentlichen Ereignissen oder beim Wechsel zwischen einer wortwörtlichen Transkription und einer zusammenfassenden Beschreibung. In der Spalte „Aussage Kind“ sind die wortwörtlichen Meldungen des jeweiligen Kindes, in der Spalte „Aussage TL“ sind die wortwörtlichen Meldungen der Testleitung erfasst. In der Spalte „Beobachtung“ ist eine Zusammenfassung des Geschehens beschrieben. Zum Erfassen spezieller Handlungen und Aktionen dient für das Kind die Spalte „Aktion Kind“, für die Testleitung die Spalte „Aktion TL“.

Zusätzlich markiert die Verfasserin der gegenständlichen Arbeit in diesen Transkriptions-Dateien das Auftreten einer Frage- bzw. Aufgabenstellung und das Beantworten bzw. Lösen einer Frage- bzw. Aufgabenstellung und kodiert die Hilfestellungen durch ein „H“ in der betreffenden Zeile.

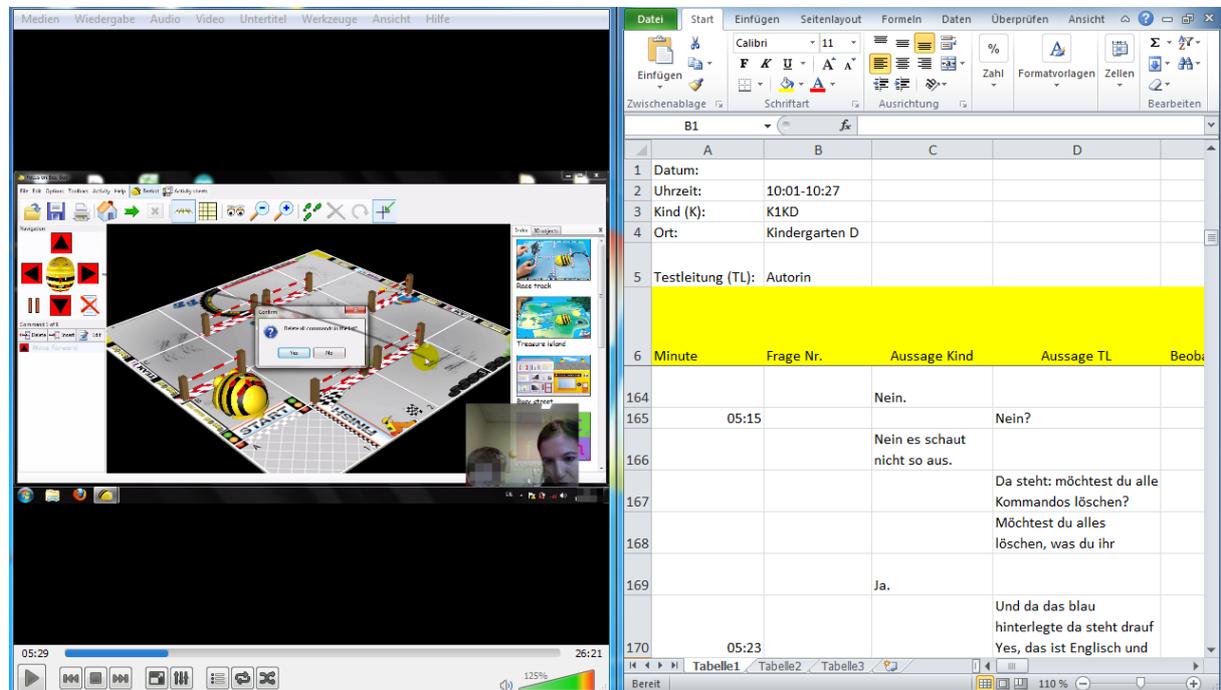


Abbildung 29: Transkription Software Usability Test

9.2.5 Kategoriensystem zur Kodierung

Die Autorin entwickelte die Kategorien für die Kodierung der Inhalte zur qualitativen Inhaltsanalyse entlang der Leitfragen aus den Transkripten. Dazu suchte die Verfasserin der

gegenständlichen Arbeit für jede Frage aus der Menge der Software Usability Tests in den Transkriptionen der jeweiligen Tests nach interessanten Aspekten, die entlang der Forschungsfragen ausgewertet werden können. Diese induktive Kategorienfindung (siehe Kapitel 7.2.3.1) erschien notwendig, da gänzlich neues Wissen aus den Software Usability Tests gewonnen werden sollte. Bei der Kategorienfindung berücksichtigte die Autorin nicht nur, dass zwei Sets an Fragestellungen vorlagen (siehe Kapitel 9.2.3), sondern auch dass die Evaluierung von zwei Versionen der Software Focus on Bee-Bot stattfand. So entstanden in Summe 288 Kategorien, zugeordnet zu den Fragestellungen.

Nach dem die Anzahl an Kategorien deutlich zu hoch war und die Verfasserin der gegenständlichen Arbeit auch nicht ausschließen konnte, dass manche redundant waren, wurde die Quantität der Kategorien reduziert. Dazu bildete die Autorin in mehreren Iterationen 11 Hauptkategorien, die thematisch zugeordnete Unterkategorien beinhalteten, nutzte zur erneuten Kategorienzuordnung ein Closed Card Sorting Verfahren und ordnete die bestehenden 288 Kategorien den finalen 11 Hauptkategorien zu. Wie in Abbildung 30 dargestellt, hat die Autorin die einzelnen Kategorien (später Unterkategorien) als Papierstreifen aufbereitet und anschließend den 11 Hauptkategorien (jeweils in einer Klarsichtfolie) zugeordnet.



Abbildung 30: Iteratives Closed Card Sorting Verfahren zur Hauptkategorienbildung, vorletzte Iteration mit 12 Hauptkategorien

Nach der Zuordnung konsolidierte die Verfasserin der gegenständlichen Arbeit pro Hauptkategorie die jeweiligen Unterkategorien, fasste ähnlich lautende zusammen, entfernte redundante und entwickelte falls notwendig neue Unterkategorien. Dies jedoch immer unter der Prämisse, dass einer neuen Unterkategorie stets zumindest zwei der bestehenden Unterkategorien zugeordnet werden können. Im Vordergrund stand die Reduktion der Anzahl an Unterkategorien bei gleichbleibender oder steigender inhaltlicher Qualität. So entstand ein finales Kategoriensystem mit 11 Hauptkategorien und 105 Unterkategorien, die in weiterer Folge zu kodieren waren.

SW01	Vorwissen
SW0101	Kind gibt an, schon einmal Computer gespielt / genutzt zu haben
SW0102	Kind gibt an, gemeinsam mit einem Elternteil den Computer genutzt zu haben
SW0103	Kind gibt an, schon einmal so ein ähnliches Programm gesehen zu haben
SW0104	Kind erwähnt, dass ein rotes Kreuz immer löschen bedeutet
SW0105	Kind gibt an, schon einmal eine Computermaus bedient zu haben
SW0106	Kind ist mit dem Konzept des Auswählens eines Elements durch Klick mit der linken Maustaste vertraut
SW02	Wahrnehmung
SW0201	Kind erkennt im Bee-Bot eine Biene
SW0202	Kind glaubt / thematisiert, dass der Bee-Bot ein Roboter ist
SW0203	Kind glaubt / thematisiert, dass der Bee-Bot kein Roboter ist, weil er ein Tier ist
SW0204	Kind erkennt im Bee-Bot ein anderes Tier
SW0205	Kind interpretiert, dass der Bee-Bot in der Software fahren kann
SW0206	Kind interpretiert, dass der Bee-Bot in der Software springen kann

SW0207	Kind interpretiert, dass der Bee-Bot in der Software fliegen kann
SW0208	Kind interpretiert, dass der Bee-Bot in der Software über Hürden springen, fliegen, ... kann
SW0209	Kind erkennt / thematisiert, dass ein eingespeicherter (Richtungs-)Befehl in der Befehlsliste erscheint
SW0210	Kind glaubt, dass der Constructa-Bot ein Roboter ist
SW0211	Kind erkennt im Constructa-Bot ein Baugerät
SW0212	Kind nennt zu Beginn des Testsettings nur Details von / auf der Matte
SW03	Programmlogik / Benutzeroberfläche: Bedienung Software
SW0301	Kind interpretiert, dass das Programm mit der Computermouse gesteuert wird
SW0302	Kind interpretiert, dass das Programm mit anderen Eingabegeräten (Tastatur, TouchPad, TouchScreen, Lenkrad, Fernsteuerung) zu bedienen ist
SW0303	Kind interpretiert, dass das Programm mit Spracheingabe zu bedienen ist
SW0304	Kind hat keine Probleme Clear- und Pause-Button zu unterscheiden
SW0305	Kind interpretiert Links- und Rechts-Dreh-Buttons als 90 Grad Drehung und nicht als Schritt nach links / rechts
SW0306	Kind interpretiert GO-Button im Navigationsbereich als „Start“ für das Losfahren des Bee-Bots
SW0307	Kind interpretiert GO-Button als Vorwärts-Befehl
SW0308	Kind interpretiert, dass der Bee-Bot auf der Matte auf seinem Rücken bedient / gesteuert wird
SW0309	Kind erkennt Navigationsbereich ohne Hilfestellung
SW0310	Kind erkennt Navigationsbereich mit Hilfestellung
SW0311	Kind interpretiert, dass der Bee-Bot im Navigationsbereich auf dem Rücken bedient werden kann
SW0312	Kind interpretiert, dass der Bee-Bot durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung gesteuert werden kann
SW0313	Kind findet selbst den GO-Button bzw. grünen Pfeil in der Symbolleiste als Befehl für das Losfahren des Bee-Bots
SW0314	Kind findet Button für Clear auf Anhieb mit max. einem Hinweis
SW0315	Kind interpretiert beliebigen Button (außer Clear selbst) als Clear-Button
SW0316	Kind findet Button für „Einzelschritte anzeigen“ auf Anhieb mit max. einem Hinweis
SW0317	Kind interpretiert beliebigen Button (außer „Einzelschritte anzeigen“ selbst) als Button für „Einzelschritte anzeigen“
SW0318	Kind interpretiert, dass der Bee-Bot durch beliebige Mausinteraktion (z.B. Drag-&-Drop oder Bee-Bot schieben, ...) gesteuert werden kann
SW0319	Kind interpretiert, dass Bee-Bot durch Klicken auf einen Befehl in der Befehlsliste gesteuert werden kann
SW0320	Kind benötigt Hilfestellung zur Eingabe einer geeigneten vollständigen Befehlsabfolge (zur Lösung einer Aufgabe)
SW0321	Kind interpretiert beliebige Position auf der Oberfläche (außer GO) für den GO-Button
SW0322	Kind findet es gut oder ist nicht irritiert, dass der Bee-Bot immer von seinem Startfeld aus beginnt alle eingespeicherten Befehle abzuarbeiten
SW0323	Kind kann auftretende Pop-Ups lesen, sinngemäß interpretieren und entsprechend handeln
SW0324	Kind thematisiert, dass der Bee-Bot nach Drücken eines Richtungs-Buttons nicht losfährt
SW04	Sinngemäße Bedienung
SW0401	Kind interpretiert Funktionalität von GO sinngemäß (Abarbeitung der gespeicherten Befehle, bis der Speicher mit Clear gelöscht wird)
SW0402	Kind nutzt bei der Focus on Bee-Bot 2 Software häufiger den GO-Button im Navigationsbereich als den in der Symbolleiste
SW0403	Kind speichert eine für die Aufgabenstellung geeignete Abfolge von Richtungsbefehlen ohne Hilfestellung ein
SW0404	Kind bringt meistens nach Eingabe einer für die Aufgabenstellung geeigneten Befehlsabfolge den Bee-Bot ohne Hilfestellung zum Fahren
SW0405	Kind erkennt, dass der von ihm eingespeicherte Weg nicht der kürzeste / schnellste war
SW0406	Kind erkennt Hindernis als unüberwindbares Hindernis
SW0407	Kind erkennt Hindernis als unüberwindbares Hindernis und erkennt, dass es umfahren werden muss
SW0408	Kind erkennt Hindernis als unüberwindbares Hindernis und umfährt es
SW0409	Kind versteht, was es bedeutet „zu löschen“
SW0410	Kind erkennt / thematisiert Unterschiede zwischen aktiven und nicht aktiven Buttons

SW0411	Kind ist der Meinung, dass der Bee-Bot „vergessen“ kann
SW0412	Kind thematisiert, dass Befehle aneinandergereiht werden
SW0413	Kind simuliert „Einzelschritte anzeigen“ durch Betätigen des GO-Buttons nach (fast) jeder (Richtungs-)Befehls-Eingabe
SW0414	Kind glaubt, den GO-Button so oft klicken zu müssen, wie Richtungs-Buttons gedrückt wurden
SW0415	Kind glaubt, dass bei der Verwendung von „Einzelschritte anzeigen“ das anschließende Betätigen des GO-Buttons nichts bewirkt
SW0416	Kind scheint nach Erklärung die Funktionalität „Einzelschritte anzeigen“ zu verstehen
SW0417	Kind ist irritiert durch Wechsel zwischen den Modi „Einzelschritte an“, „Einzelschritte aus“
SW05	Orientierung
SW0501	Kind gleicht durch Rotieren der Matte unterschiedliche Perspektiven zwischen dem Kind und dem Bee-Bot auf der Matte im Programm aus (oder behilft sich durch Drehen leichtere Perspektive zu finden)
SW0502	Kind bedient Richtungs-Buttons trotz anderer Blickrichtung des Bee-Bots in Relation zum Navigationsbereich korrekt
SW0503	Es scheint, dass dem Kind die Bedienung des Programms leichter fällt, wenn der Bee-Bot in Relation zum Navigationsfeld in die gleiche Richtung sieht
SW0504	Kind hat Probleme mit der Unterscheidung / Umsetzung von Links- und Rechts-Dreh-Befehlen
SW0505	Kind kann Richtungsfehler erkennen und selbst korrigieren
SW06	Haptik
SW0601	Kind findet nach Abrutschen den Maus-Cursor am Bildschirm nicht mehr
SW0602	Kind merkt nicht, dass es beim Mausklick auf einen Button abgerutscht ist und der Befehl nicht gespeichert / ausgeführt wurde
SW0603	Kind kommt näher zum Bildschirm, um den Inhalt besser erkennen zu können
SW0604	Kind hat Probleme bei der Nutzung der Computermaus (Positionierung, Tasten erreichen, klicken, ...)
SW0605	Kind fährt mit der Computermaus ohne Umwege zum Zielobjekt
SW07	Allgemeine Fehler
SW0701	Kind dreht unabsichtlich die Matte
SW0702	Kind kann die Matte nicht in die Ausgangsposition drehen
SW0703	Kind erkennt nicht, dass durch das Programm unausführbare Befehle (z.B. wegen Hürden) automatisch gelöscht wurden
SW0704	Kind kann das Drehen der Matte selbständig beenden
SW08	Unterschiede
SW0801	Kind erkennt, dass es in der Focus on Bee-Bot 2 Software 2 Möglichkeiten für GO gibt
SW0802	Kind findet Matten in folgender Software schöner / besser
SW0803	Kind empfindet folgende Focus on Bee-Bot Software leichter (zur Bedienung)
SW0804	Kind empfindet folgenden Navigationsbereich als leichter
SW0805	Kind thematisiert Unterschied in den Links- und Rechts-Dreh-Buttons der beiden Software-Versionen
SW0806	Kind gefällt folgende Focus on Bee-Bot Software besser, ohne Angabe von Gründen
SW0807	Kind findet bei folgender Focus on Bee-Bot Software die Steuerung / Bedienung besser
SW0808	Kind empfindet die optische Aufbereitung bei folgender Software ansprechender
SW09	Wissenstransfer
SW0901	Kind erkennt, dass die Matten grundsätzlich so aussehen, wie die in Natura zur Verfügung stehenden
SW0902	Kind thematisiert Unterschiede zum realen Bee-Bot
SW0903	Kind thematisiert, dass der reale Bee-Bot stechen kann (böse ist)
SW0904	Kind erkennt Bee-Bot in der Software als Bee-Bot wieder
SW0905	Kind erkennt Unterschiede bei den Buttons zwischen realem Bee-Bot und dem Navigationsfeld in der Software
SW0906	Kind erkennt Unterschiede bei den Buttons zwischen den beiden Software-Versionen
SW0907	Kind gibt an, durch die jeweils aufgefundenen Unterschiede irritiert zu sein
SW0908	Kind kann Erfahrung / Wissen aus der Nutzung der Focus on Bee-Bot 1 Software in die Nutzung der Focus on Bee-Bot 2 Software übertragen
SW0909	Kind fällt es schwer, die Lösung einer Aufgabe aus der realen Welt in die virtuelle Welt zu übertragen
SW0910	Kind kann Erfahrung / Wissen aus der Nutzung der Hardware in die Nutzung der Software übertragen

SW0911	Kind kann Erfahrung / Wissen aus der Nutzung des Bee-Bots, der Software Focus on Bee-Bot 1 bzw. 2 auf den Constructa-Bot übertragen
SW0912	Kind thematisiert das Zurückstellen des Bee-Bots auf das Startfeld, wenn der GO-Button gedrückt wird und der Bee-Bot an einer anderen Position gestanden hat, als auf dem Startfeld, als Unterschied zum realen Bee-Bot
SW0913	Kind thematisiert bei Focus on Bee-Bot, dass Buttons ausgegraut sind (im Moment nicht verfügbar) als Unterschied zum realen Bee-Bot
SW10	Präferenzen
SW1001	Kind empfindet die Funktion „Einzelschritte anzeigen“ als hilfreich
SW1002	Kind empfindet Bee-Bot ansprechender als Constructa-Bot
SW1003	Kind gefällt Constructa-Bot besser (weil lustiger, weil er transportieren kann, weil er zeichnen kann, etc.)
SW1004	Kind bevorzugt Software gegenüber realem Bee-Bot
SW11	Testsetting
SW1101	Kind stellt sich selbst (weitere) Aufgabenstellungen mit dem Bee-Bot
SW1102	Kind kommentiert eigenes Handeln
SW1103	Kind stellt von sich aus Fragen an TL
SW1104	Kind interagiert mit TL (antwortet auf Nachfragen)
SW1105	Kind ignoriert Aufgabenstellung und klickt auf der Oberfläche
SW1106	Kind lässt sich durch andere Programmelemente während einer Aufgabe ablenken
SW1107	Kind meint, dass es die Aufgabe nicht lösen kann

Tabelle 16: Kategoriensystem Software Usability Test, finale Version

Die Hinweise zur Kodierung des Kategoriensystems sind in Anhang A5 zu finden.

9.2.6 Ergebnisse

Die Autorin kodierte die 18 Transkriptions-Dateien, wobei sie im Verlauf der Kodierung erkannte, dass ein Test im SWT2, Kindergarten E, nicht vollständig kodiert werden konnte, da das Kind zu wenig mit der Testleitung interagierte und zu schüchtern war. Die Kodierung dieses Tests ist nicht in den Ergebnissen enthalten.

Bei einem weiteren Kind (ebenfalls im SWT2, Kindergarten E) musste die Evaluierung der Software Usability auf Grund mangelnder Motivation des Kindes abgebrochen werden. Bis zu diesem Zeitpunkt jedoch, konnte der Test kodiert werden und ist daher in den Ergebnissen enthalten.

Die Ergebnisse der Kodierung des Software Usability Tests werden entlang des Kategoriensystems dargestellt.

9.2.6.1 SW01 Vorwissen

9.2.6.1.1 SWT1

Von 9 im SWT1 befragten Kindern gaben 8 Kinder an, schon einmal einen Computer genutzt / mit einem Computer gespielt zu haben (SW0101), davon gaben 2 Kinder an, dass dies gemeinsam mit einem Elternteil (SW0102) geschah. Nach Angaben der Kinder hatte keines von ihnen schon einmal ein ähnliches Programm gesehen (SW0103). 2 der im SWT1 befragten 9 Kinder erwähnten, dass ein rotes Kreuz immer löschen bedeutet (SW0104). 4 der im SWT1 befragten Kinder gaben an, schon einmal eine Computermaus bedient zu haben (SW0105). Mit dem Konzept des Auswählens eines Elements mit der linken Maustaste (SW0106) waren von diesen Kindern 2 vertraut. 2 weitere Kinder waren ebenfalls mit diesem Konzept vertraut (SW0106) ohne dass sie eine frühere Computermausnutzung erwähnt hatten (SW0105).

9.2.6.1.2 SWT2

4 der 8 beim SWT2 befragten Kinder gaben an, schon einmal einen Computer genutzt / mit einem Computer gespielt zu haben (SW0101), wobei keines der Kinder eine gemeinsame Nutzung mit einem Elternteil erwähnte (SW0102). Ein Kind gab an, schon einmal ein

ähnliches Programm gesehen zu haben (SW0103), kein Kind erwähnte, dass ein rotes Kreuz immer löschen bedeutet (SW0104). 2 der im SWT2 befragten Kinder gaben an, schon einmal eine Computermaus bedient zu haben (SW0105), beide waren mit dem Konzept des Auswählens eines Elements mit der linken Maustaste vertraut (SW0106). Darüber hinaus waren weitere 3 Kinder mit diesem Konzept vertraut (SW0106), wobei 2 dieser Kinder zuvor angegeben hatten, schon einmal einen Computer genutzt / mit einem Computer gespielt zu haben (SW0101).

9.2.6.2 SW02 Wahrnehmung

9.2.6.2.1 *SWT1*

Alle im SWT1 befragten Kinder erkannten im Bee-Bot im Programm eine Biene (SW0201), wobei ein Kind zuerst der Meinung war, dass es sich um ein anderes Tier handeln würde (SW0204) und dann die Meinung revidierte. Aufgrund des Vorwissens der Kinder über den realen Bee-Bot wurden die Fragen zu den Kategorien SW0202 und SW0203 im SWT1 nicht gestellt. Alle im SWT1 befragten Kinder erkannten, dass der Bee-Bot in der Software fahren kann (SW0205). Darüber hinaus interpretierten 4 Kinder, dass der Bee-Bot in der Software springen kann (SW0206) und von diesen äußerten 2 Kinder überdies, dass der Bee-Bot in der Software fliegen kann (SW0207). Zusätzlich zu den 4 Kindern, die interpretierten, dass der Bee-Bot in der Software springen kann (SW0206), ging bei der konkreten Hürdenaufgabe noch ein zusätzliches fünftes Kind davon aus, dass der Bee-Bot in der Software über Hürden springen, fliegen,... kann (SW0208). Ein Kind erkannte / thematisierte, dass ein eingespeicherter (Richtungs-)Befehl in der Befehlsliste erschien (SW0209). 5 der 6 zu dieser Kategorie im SWT1 befragten Kinder glaubten, dass der Constructa-Bot ein Roboter ist (SW0210), ein Kind war anderer Meinung. 6 der 9 Kinder, die den Constructa-Bot gesehen hatten, identifizierten ihn als Baugerät (SW0211), darunter das eine Kind, das nicht glaubte, dass der Constructa-Bot ein Roboter ist (SW0210). 2 Kinder, die äußerten, dass der Constructa-Bot ein Roboter ist (SW0210), erkannten ihn nicht als Baugerät (SW0211). Alle 9 im SWT1 befragten Kinder nannten zu Beginn des Testsettings nur Details von / auf der Matte (in der Mitte) des Programms (SW0212).

9.2.6.2.2 *SWT2*

3 der 8 im SWT2 befragten Kinder erkannten im Bee-Bot im Programm eine Biene (SW0201), 5 Kinder, davon eines, das die Meinung auf Biene (SW0201) änderte, sahen im Bee-Bot ein anderes Tier (SW0204), ein Kind erkannte kein Tier im Bee-Bot. 4 Kinder, davon ein Kind, das im Bee-Bot eine Biene erkannte (SW0201), glaubten / thematisierten, dass der Bee-Bot ein Roboter ist (SW0202). 2 Kinder, davon eines, das glaubte, dass es sich beim Bee-Bot um eine Biene handelt (SW0201), waren der Meinung, dass es sich beim Bee-Bot um keinen Roboter handelt, da er ein Tier ist (SW0203). 4 der 7 zu dieser Kategorie im SWT2 befragten Kinder interpretierten, dass der Bee-Bot in der Software fahren kann (SW0205), 2 von diesen und 2 weitere der 7 zu dieser Kategorie im SWT2 befragten Kinder interpretierten, dass der Bee-Bot in der Software fliegen kann (SW0207). Von den 7 zu dieser Kategorie im SWT2 befragten Kindern, interpretierten 2 Kinder, dass der Bee-Bot in der Software springen kann (SW0206). Eines dieser Kinder sah alle 3 kodierten Fortbewegungsmöglichkeiten für den Bee-Bot in der Software (SW0205, SW0206, SW0207), und das andere Kind gab springen (SW0206) von den 3 kodierten als einzige Fortbewegungsmöglichkeit des Bee-Bots im Programm an. Diese 2 Kinder interpretierten bei der konkreten Hürdenaufgabe auch, dass der Bee-Bot in der Software über Hürden springen, fliegen,... kann (SW0208). 3 der 8 an der Usability-Evaluierung im SWT2 teilnehmenden Kinder erkannten / thematisierten, dass ein eingespeicherter (Richtungs-)Befehl in der Befehlsliste erschien (SW0209). Die Fragen zu den Kategorien SW0210 und SW0211 wurden aufgrund der (absichtlich) fehlenden Vorerfahrung der befragten Kinder mit dem realen Bee-Bot im SWT2 nicht gestellt. Alle 8 befragten Kinder nannten zu Beginn des Testsettings nur Details von / auf der Matte (in der Mitte) des Programms (SW0212).

9.2.6.3 SW03 Programmlogik / Benutzeroberfläche: Bedienung Software

9.2.6.3.1 SWT1

8 der 9 im SWT1 befragten Kinder interpretierten, dass das Programm mit der Computermaus gesteuert wird (SW0301). Bevor sie die Computermaus für die Steuerung identifizierten (SW0301), gingen 2 dieser Kinder davon aus, dass das Programm mit anderen Eingabegeräten wie Tastatur, TouchPad, TouchScreen, Lenkrad oder Fernsteuerung zu bedienen ist (SW0302), dies interpretierten zu verschiedenen Zeitpunkten im Testsetting auch 4 weitere Kinder (SW0302). Ein Kind versuchte im Laufe des SWT1 das Programm mit Spracheingabe zu bedienen (SW0303). 5 Kinder interpretierten im Laufe des Testsettings, dass der Bee-Bot auf der Matte im Programm auf seinem Rücken bedient / gesteuert wird (SW0308), 4 dieser Kinder und 3 weitere vermuteten, dass der Bee-Bot im Navigationsbereich der Software Focus on Bee-Bot 1 auf dem Rücken, auf dem er aufgemalte Richtungstasten aufweist, bedient werden kann (SW0311). 2 Kinder, die interpretierten, dass der Bee-Bot auf der Matte in der Software auf seinem Rücken bedient werden kann (SW0308) vermuteten auch, dass der Bee-Bot im Programm durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung gesteuert werden kann (SW0312). Ein Kind, das interpretierte, dass der Bee-Bot im Programm auf seinem Rücken bedient / gesteuert wird (SW0308) und dass der Bee-Bot im Navigationsbereich der Software Focus on Bee-Bot 1 auf dem Rücken bedient werden kann (SW0311), versuchte den Bee-Bot durch beliebige Mausinteraktion wie z.B. Drag-&Drop oder Schieben zu steuern (SW0318). Ein Kind interpretierte in der Software Focus on Bee-Bot 1, dass der Bee-Bot durch Klicken auf einen Befehl in der Befehlsliste gesteuert werden kann (SW0319). Kein Kind erkannte im SWT1 den Navigationsbereich ohne Hilfestellung (SW0309), mit Hilfestellung erkannten diesen 8 Kinder (SW0310).

4 Kinder thematisierten, dass der Bee-Bot nach Drücken eines Richtungs-Buttons nicht losfuhr (SW0324), eines dieser Kinder fand selbst den GO-Button bzw. grünen Pfeil in der Symbolleiste der Software Focus on Bee-Bot 1 als Befehl für das Losfahren des Bee-Bots (SW0313) und benötigte keine Hilfestellung zur Eingabe einer geeigneten vollständigen Befehlsabfolge (zur Lösung einer Aufgabe) (SW0320). In der Software Focus on Bee-Bot 1 interpretierten 8 Kinder eine beliebige Position auf der Oberfläche (außer GO) als GO-Button (SW0321), ein Kind davon interpretierte in der Software Focus on Bee-Bot 2 eine beliebige Position auf der Oberfläche (außer GO) als GO-Button (SW0321). Dieses Kind und ein weiteres deuteten in der Software Focus on Bee-Bot 1 den GO-Button als Vorwärts-Befehl (SW0307). 8 Kinder sahen den GO-Button im Navigationsbereich der Focus on Bee-Bot 2 Software als „Start“ für das Losfahren des Bee-Bots an (SW0306). 5 Kinder empfanden es als gut, oder waren nicht irritiert, dass der Bee-Bot immer von seinem Startfeld aus beginnt alle eingespeicherten Befehle abzuarbeiten (SW0322). 7 Kinder fanden im SWT1 auf Anhieb mit maximal einem Hinweis den Button für Clear (SW0314), ein Kind von diesen und 2 weitere interpretierten einen beliebigen Button (außer Clear selbst) als Clear-Button (SW0315).

Im Laufe des SWT1 hatten (nach erstem Auffinden von Clear) bei der Software Focus on Bee-Bot 1 5 Kinder keine Probleme Clear- und Pause-Button zu unterscheiden, bei 2 Kindern trat in dieser Software Version kein weiterer Löschvorgang auf, Probleme diesbezüglich hatten 2 Kinder (SW0304). Bei der Software Focus on Bee-Bot 2 hatten während des SWT1 (nach erstem Auffinden von Clear) 6 Kinder mit dem Unterscheiden der Buttons für Clear und Pause Probleme, 2 Kinder hatten keine diesbezüglichen Probleme und bei einem Kind kam es zu keinem weiteren Löschvorgang in dieser Software Version (SW0304). Ein Kind konnte auftretende Pop-Ups lesen, sinngemäß interpretieren und entsprechend handeln, bei 8 Kindern war dies ohne Hilfe nicht der Fall (SW0323). 5 der 9 an der Usability Evaluierung SWT1 teilnehmenden Kinder interpretierten die Links- und Rechts-Dreh-Buttons als 90 Grad Drehung und nicht als Schritt nach links bzw. rechts, 4 Kinder interpretierten Schritte nach links bzw. rechts anstelle der Drehung (SW0305). 4 Kinder fanden den Button für die Funktion „Einzelschritte anzeigen“ auf Anhieb mit maximal einem

Hinweis (SW0316), 4 andere Kinder interpretierten einen beliebigen Button (außer „Einzelschritte anzeigen“ selbst) als Button für „Einzelschritte anzeigen“ (SW0317).

9.2.6.3.2 SWT2

6 der 8 an der Usability Evaluierung SWT2 teilnehmenden Kinder interpretierten, dass das Programm mit der Computermouse gesteuert wird (SW0301). 2 Kinder deuteten, dass das Programm mit anderen Eingabegeräten wie Tastatur, TouchPad, TouchScreen, Lenkrad oder Fernsteuerung zu bedienen ist (SW0302). Diese beiden Kinder und 3 weitere versuchten im Laufe des Testsettings, das Programm mit Spracheingabe zu bedienen (SW0303). 5 Kinder interpretierten im Laufe des Testsettings, dass der Bee-Bot auf der Matte im Programm auf seinem Rücken bedient / gesteuert wird (SW0308), 3 dieser Kinder deuteten, dass der Bee-Bot im Navigationsbereich der Software Focus on Bee-Bot 1 auf dem Rücken, auf dem er aufgemalte Richtungstasten aufweist, bedient werden kann (SW0311). 2 Kinder, die interpretierten, dass der Bee-Bot auf der Matte in der Software auf seinem Rücken bedient werden kann (SW0308) versuchten im Laufe des Testsettings den Bee-Bot im Programm durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung zu steuern (SW0312), was auch ein weiteres Kind so deutete (SW0312). Ein Kind, das interpretierte, dass der Bee-Bot im Programm auf seinem Rücken bedient / gesteuert wird (SW0308), dass der Bee-Bot im Navigationsbereich der Software Focus on Bee-Bot 1 auf dem Rücken bedient werden kann (SW0311) und dass der Bee-Bot durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung gesteuert werden kann (SW0312) interpretierte, dass der Bee-Bot durch beliebige Mausinteraktionen wie z.B. Drag-&-Drop oder schieben des Bee-Bots gesteuert werden kann (SW0318). Dieses Kind und ein weiteres deuteten in der Software Focus on Bee-Bot 1, dass der Bee-Bot durch Klicken auf einen Befehl in der Befehlsliste gesteuert werden kann (SW0319).

Ein Kind erkannte den Navigationsbereich ohne Hilfestellung (SW0309), mit Hilfestellung erkannten diesen 6 Kinder von selbst (SW310). Das Kind, das den Navigationsbereich ohne Hilfestellung gefunden hatte (SW0309), thematisierte, dass der Bee-Bot nach Drücken eines Richtungs-Buttons nicht losfuhr (SW0324). Kein Kind fand selbst den GO-Button bzw. grünen Pfeil in der Symbolleiste der Software Focus on Bee-Bot 1 als Befehl für das Losfahren des Bee-Bots (SW0313) und alle Kinder benötigten Hilfestellung zur Eingabe einer geeigneten vollständigen Befehlsabfolge (zur Lösung einer Aufgabe) (SW0320). In der Software Focus on Bee-Bot 1 interpretierten beim SWT2 5 Kinder eine beliebige Position auf der Oberfläche (außer GO) als GO-Button (SW0321), 2 dieser Kinder und ein weiteres interpretierten in der Software Focus on Bee-Bot 2 eine beliebige Position auf der Oberfläche (außer GO) als GO-Button (SW0321). 2 Kinder, die in der Software Focus on Bee-Bot 1 eine beliebige Position auf der Oberfläche (außer GO) für den GO-Button hielten (SW0321), deuteten in dieser Software Version den GO-Button als Vorwärts-Befehl (SW0307). 3 Kinder interpretierten den GO-Button im Navigationsbereich der Focus on Bee-Bot 2 Software als „Start“ für das Losfahren des Bee-Bots (SW0306). 7 Kinder fanden es gut, oder waren nicht irritiert, dass der Bee-Bot immer von seinem Startfeld aus beginnt alle eingespeicherten Befehle abzuarbeiten (SW0322). 2 Kinder fanden auf Anhieb mit maximal einem Hinweis den Button für Clear (SW0314), 3 andere Kinder interpretierten einen beliebigen Button (außer Clear selbst) als Clear-Button (SW0315).

Im Laufe des Testsettings hatten (nach erstem Auffinden von Clear) bei der Focus on Bee-Bot 1 Software 4 Kinder keine Probleme Clear- und Pause-Button zu unterscheiden (SW0304), bei 4 Kindern trat in dieser Software Version kein weiterer Löschvorgang auf (SW0304). Bei der Software Focus on Bee-Bot 2 hatten im Laufe des SWT2 (nach erstem Auffinden von Clear) 5 Kinder keine Probleme mit dem Unterscheiden der Buttons für Clear und Pause, ein Kind hatte diesbezüglich Probleme und bei 2 weiteren Kindern kam es in dieser Software Version zu keinem weiteren Löschvorgang (SW0304). Kein Kind konnte im SWT2 auftretende Pop-Ups lesen, sinngemäß interpretieren und entsprechend handeln (SW0323). Ein Kind interpretierte die Links- und Rechts-Dreh-Buttons als 90 Grad Drehung und nicht als Schritt nach links bzw. rechts, 6 Kinder interpretierten Schritte nach links bzw.

rechts anstelle der Drehung (SW0305). 2 Kinder fanden den Button für die Funktion „Einzelschritte anzeigen“ auf Anhieb mit maximal einem Hinweis (SW0316), ein anderes Kind interpretierte einen beliebigen Button (außer „Einzelschritte anzeigen“ selbst) als Button für „Einzelschritte anzeigen“ (SW0317).

9.2.6.4 SW04 Sinngemäße Bedienung

9.2.6.4.1 SWT1

6 der 9 an der Usability Evaluierung SWT1 teilnehmenden Kinder interpretierten die Funktionalität von GO sinngemäß als Abarbeiten der gespeicherten Befehle, bis der Speicher mit Clear gelöscht wird (SW0401). 7 der Kinder nutzten bei der Software Focus on Bee-Bot 2 häufiger den GO-Button im Navigationsbereich als den in der Symbolleiste (SW0402). 5 Kinder speicherten im SWT1 eine für die Aufgabenstellung geeignete Abfolge von Richtungsbefehlen ohne Hilfestellung ein (SW0403) und 2 Kinder brachten den Bee-Bot meistens nach Eingabe einer für die Aufgabenstellung geeigneten Befehlsabfolge ohne Hilfestellung zum Fahren (SW0404). Ein Kind erkannte, dass der von ihm eingespeicherte Weg nicht der kürzeste / schnellste war (SW0405). 6 Kinder nahmen das Hindernis in der Focus on Bee-Bot 1 Software (Hürden) als unüberwindbares Hindernis wahr (SW0406), 5 dieser Kinder und 2 weitere erkannten, dass dieses Hindernis umfahren werden musste (SW0407). 5 dieser 7 Kinder umfuhren das Hindernis (SW0408). 4 der 7 Kinder, die später interpretierten, dass das Hindernis umfahren werden muss (SW0407), interpretierten beim Öffnen der Software, dass der Bee-Bot im Programm nicht über Hürden springen, fliegen oder ähnliches kann (SW0208).

Alle 9 Kinder im SWT1 waren vor dem ersten Löschvorgang der Meinung, dass der Bee-Bot im Programm „vergessen“ kann (SW0411), 8 dieser Kinder verstanden auch, was es bedeutet „zu löschen“ (SW0409). 2 Kinder thematisierten, dass Befehle aneinandergereiht wurden (SW0412). 4 Kinder erkannten / thematisierten Unterschiede zwischen aktiven und nicht aktiven Buttons (SW0410). 6 Kinder simulierten die Funktion „Einzelschritte anzeigen“ durch Betätigen des GO-Buttons nach (fast) jeder (Richtungs-)Befehls-Eingabe (SW0413), eines dieser Kinder glaubte, den GO-Button so oft klicken zu müssen, wie Richtungs-Buttons gedrückt wurden (SW0414). 3 Kinder waren im SWT1 durch den Wechsel zwischen den Modi „Einzelschritte an“, „Einzelschritte aus“ irritiert (SW0417), 4 andere Kinder schienen nach Erklärung die Funktionalität „Einzelschritte anzeigen“ zu verstehen (SW0416). Kein Kind glaubte, dass bei der Verwendung der Funktion „Einzelschritte anzeigen“ das anschließende Betätigen des GO-Buttons nichts bewirkt (SW0415).

9.2.6.4.2 SWT2

Eines der an der Usability Evaluierung SWT2 teilnehmenden Kinder interpretierte die Funktionalität von GO sinngemäß als Abarbeiten der gespeicherten Befehle, bis der Speicher mit Clear gelöscht wird (SW0401). 2 der Kinder nutzten bei der Software Focus on Bee-Bot 2 häufiger den GO-Button im Navigationsbereich als den in der Symbolleiste (SW0402). Kein Kind speicherte eine für die Aufgabenstellung geeignete Abfolge von Richtungsbefehlen ohne Hilfestellung ein (SW0403) und kein Kind brachte meistens nach Eingabe einer für die Aufgabenstellung geeigneten Befehlsabfolge den Bee-Bot ohne Hilfestellung zum Fahren (SW0404). Ein Kind erkannte, dass der von ihm eingespeicherte Weg nicht der kürzeste / schnellste war (SW0405). 6 Kinder nahmen im SWT2 das Hindernis in der Focus on Bee-Bot 1 Software (Hürden) als unüberwindbares Hindernis wahr (SW0406), davon erkannten 4 Kinder und ein weiteres Kind, dass das Hindernis umfahren werden musste (SW0407). 4 dieser Kinder interpretierten beim Öffnen der Software, dass der Bee-Bot im Programm nicht über Hürden springen, fliegen oder ähnliches kann (SW0208). Ein Kind umfuhr das Hindernis (SW0408). 5 Kinder waren vor dem ersten Löschvorgang der Meinung, dass der Bee-Bot im Programm „vergessen“ kann (SW0411), diese Kinder verstanden auch, was es bedeutet „zu löschen“ (SW0409). Eines dieser Kinder thematisierte, dass Befehle aneinandergereiht wurden (SW0412).

Ein Kind erkannte / thematisierte im SWT2 Unterschiede zwischen aktiven und nicht aktiven Buttons (SW0410) und ging davon aus, den GO-Button sooft klicken zu müssen, wie Richtungs-Buttons gedrückt wurden (SW0414). 2 Kinder simulierten die Funktion „Einzelschritte anzeigen“ durch Betätigen des GO-Buttons nach (fast) jeder (Richtungs-)Befehls-Eingabe (SW0413). Ein Kind war durch den Wechsel zwischen den Modi „Einzelschritte an“, „Einzelschritte aus“ irritiert (SW0417), dieses Kind schien nach Erklärung die Funktionalität „Einzelschritte anzeigen“ zu verstehen (SW0416). Ein anderes Kind glaubte, dass bei der Verwendung der Funktion „Einzelschritte anzeigen“ das anschließende Betätigen des GO-Buttons nichts bewirkt (SW0415).

9.2.6.5 SW05 Orientierung

9.2.6.5.1 *SWT1*

2 der 9 an der Usability Evaluierung SWT1 teilnehmenden Kinder glichen durch Rotieren der Matte unterschiedliche Perspektiven zwischen dem Kind und dem Bee-Bot auf der Matte im Programm aus (SW0501), diese 2 waren Teil der 4 Kinder, die die Matte unabsichtlich gedreht hatten (SW0701) und die Funktionalität im Programm gesehen hatten. 4 Kinder bedienten im SWT1 die Richtungs-Buttons trotz anderer Blickrichtung des Bee-Bots in Relation zum Navigationsbereich korrekt (SW0502), diese 4 Kinder und ein weiteres Kind hatten keine Probleme mit der Unterscheidung / Umsetzung von Links- und Rechts-Dreh-Befehlen (SW0504). Bei allen 9 Kindern im SWT1 schien es, dass ihnen die Bedienung des Programms leichter fiel, wenn der Bee-Bot in Relation zum Navigationsfeld (und den Kindern selbst) in die gleiche Richtung sah (SW0503), wobei 7 Kinder Richtungsfehler erkennen und selbst korrigieren konnten (SW0505).

9.2.6.5.2 *SWT2*

Bei 4 Kindern der 7 zu dieser Kategorie im SWT2 beobachteten Kinder schien es, dass ihnen die Bedienung des Programms leichter fiel, wenn der Bee-Bot in Relation zum Navigationsfeld (und den Kindern selbst) in die gleiche Richtung sah (SW0503), eines dieser Kinder hatte weder Probleme mit der Unterscheidung / Umsetzung von Links- und Rechts-Dreh-Befehlen (SW0504), noch mit dem Erkennen und Korrigieren von Richtungsfehlern (SW0505). Ein anderes Kind hatte keine Probleme mit der Unterscheidung / Umsetzung von Links- und Rechts-Dreh-Befehlen (SW0504), 2 andere Kinder konnten Richtungsfehler erkennen und selbst korrigieren (SW0505). Kein Kind bediente im SWT2 die Richtungs-Buttons trotz anderer Blickrichtung des Bee-Bots in Relation zum Navigationsbereich korrekt (SW0502) und kein Kind glich durch Rotieren der Matte unterschiedliche Perspektiven zwischen dem Kind und dem Bee-Bot auf der Matte im Programm aus (SW0501), wobei 5 Kinder die Matte im Laufe des Testsettings unabsichtlich drehten (SW0701) und diese Funktionalität im Programm gesehen hatten.

9.2.6.6 SW06 Haptik

9.2.6.6.1 *SWT1*

2 der 9 Kinder, die an der Usability-Evaluierung SWT1 teilnahmen, fanden nach Abrutschen des Maus-Cursors diesen nicht mehr am Bildschirm (SW0601). Eines dieser 2 Kinder und zusätzliche 4 merkten nicht, dass sie beim Mausklick auf einem Button abgerutscht waren und der Befehl nicht gespeichert / ausgeführt wurde (SW0602). 3 dieser 5 Kinder und 3 weitere hatten Probleme bei der Nutzung der Computermaus (Positionierung, Tasten erreichen, klicken, ...) (SW0604), wovon 4 Kinder mit der Computermaus mit Umwegen zum Zielobjekt fuhren (SW0605). 6 Kinder kamen im Laufe des SWT1 näher zum Bildschirm, um den Inhalt besser erkennen zu können (SW0603).

9.2.6.6.2 SWT2

7 der 8 an der Usability-Evaluierung SWT2 partizipierenden Kinder fanden nach Abrutschen des Maus-Cursors diesen am Bildschirm (SW0601), bei einem Kind kam das Abrutschen nicht vor (SW0601). 4 Kinder merkten nicht, dass sie beim Mausklick auf einem Button abrutschten und der Befehl nicht gespeichert / ausgeführt wurde (SW0602). 3 dieser 4 Kinder und 3 weitere hatten Probleme bei der Nutzung der Computermaus (Positionierung, Tasten erreichen, klicken, ...) (SW0604), wobei 4 dieser insgesamt 6 Kinder mit der Computermaus mit Umwegen zum Zielobjekt fuhren (SW0605). Ein Kind kam im Laufe des SWT2 näher zum Bildschirm, um den Inhalt besser erkennen zu können (SW0603).

9.2.6.7 SW07 Allgemeine Fehler

9.2.6.7.1 SWT1

4 der 9 an der Usability-Evaluierung SWT1 teilnehmenden Kinder, drehten unabsichtlich die Matte (SW0701), wovon 2 das Drehen sowohl selbständig beenden (SW0704), als auch die Matte in die Ausgangsposition drehen (SW0702) konnten. Ein Kind konnte das Drehen der Matte selbständig beenden (SW0704), diese aber nicht in die Ausgangsposition drehen (SW0702), das andere Kind konnte das Drehen der Matte nicht selbständig beenden (SW0704), die Matte aber in die Ausgangsposition drehen (SW0702). Im SWT1 drehten 4 Kinder die Matte nicht (SW0701), einem Kind wurde die Matte im Zuge des Testsettings von TL gedreht (SW0701).

Ein Kind erkannte, 2 Kinder erkannten nicht, dass durch das Programm unausführbare Befehle (z.B. wegen Hürden) automatisch gelöscht wurden (SW0703). Bei 6 Kindern trat diese Situation nicht auf, entweder, weil die Kinder keine Befehle eingespeichert hatten, die zu diesem Programmverhalten geführt hätten, oder weil das Programm (insbesondere die Software Focus on Bee-Bot 1) dieses Verhalten nicht gezeigt hat.

9.2.6.7.2 SWT2

5 der 8 Kinder, die an der Usability-Evaluierung SWT2 teilnahmen, drehten unabsichtlich die Matte (SW0701), wovon 2 sowohl das Drehen selbständig beenden (SW0704), als auch die Matte in die Ausgangsposition drehen (SW0702) konnten. 2 Kinder konnten das Drehen der Matte selbständig beenden (SW0704), diese aber nicht in die Ausgangsposition drehen (SW0702). Ein Kind konnte das Drehen der Matte selbständig beenden (SW0704), die Matte musste aufgrund der nur minimal auftretenden Drehbewegung nicht in die Ausgangsposition gedreht werden (SW0702). 3 Kinder drehten die Matte nicht (SW0701).

2 Kinder erkannten im SWT2 nicht, dass durch das Programm unausführbare Befehle (z.B. wegen Hürden) automatisch gelöscht wurden (SW0703). Bei 6 Kindern trat diese Situation (SW0703) nicht auf, entweder, weil die Kinder keine Befehle eingespeichert hatten, die zu diesem Programmverhalten geführt hätten, oder weil das Programm (insbesondere die Software Focus on Bee-Bot 1) dieses Verhalten nicht gezeigt hat.

9.2.6.8 SW08 Unterschiede

9.2.6.8.1 SWT1

4 der 9 an der Usability Evaluierung SWT1 partizipierenden Kinder erkannten, dass es in der Software Focus on Bee-Bot 2 zwei Möglichkeiten für GO (2 GO-Buttons) gibt (SW0801). Eines dieser Kinder und 2 weitere thematisierten den Unterschied der Links- und Rechts-Dreh-Buttons der beiden Software-Versionen (SW0805). Bezüglich der Unterschiede der beiden Software-Versionen wurde den Kindern die offene Frage gestellt, ob ihnen eine Software besser gefallen hatte und wenn ja aus welchem Grund / aus welchen Gründen. Aus den Nennungen der Kinder ergeben sich für den SWT1 folgende Ergebnisse: 2 Kinder fanden die Matten in der Software Focus on Bee-Bot 1 schöner / besser, 3 Kinder die der Software Focus on Bee-Bot 2 (SW0802). Ein Kind empfand die Focus on Bee-Bot 1

Software als leichter zu bedienen, 4 Kinder empfanden dies für die Software Focus on Bee-Bot 2 Software (SW0803). Der Navigationsbereich wurde bei der Software Focus on Bee-Bot 1 von 2 Kindern als leichter zu bedienen genannt, 3 empfanden dies bei der Software Focus on Bee-Bot 2 (SW0804).

Ohne Angabe von Gründen gefiel 4 Kindern die Software Focus on Bee-Bot 1 besser, 5 Kindern die Software Focus on Bee-Bot 2 (SW0806). 5 dieser 9 Kinder fanden beim SWT1 bei der Software, die ihnen ohne Angabe von Gründen besser gefiel (SW0806), die Steuerung / Bedienung besser (SW0807), 3 Kinder bei der Software Focus on Bee-Bot 2 und 2 Kinder bei der Software Focus on Bee-Bot 1. 2 Kinder, welchen die Software Focus on Bee-Bot 1 ohne Angabe von Gründen besser gefiel (SW0806), fanden die Steuerung / Bedienung der Software Focus on Bee-Bot 2 besser (SW0807), eines dieser Kinder empfand die optische Aufbereitung der Software Focus on Bee-Bot 2 als besser, das andere machte bezüglich der optischen Aufbereitung der Software keine Angaben (SW0808). 2 Kinder, welchen die Software Focus on Bee-Bot 2 ohne Angabe von Gründen besser gefiel (SW0806), machten bezüglich der Steuerung / Bedienung keine Angaben, welche sie besser fanden (SW0807). Ein Kind, dem die Software Focus on Bee-Bot 1 ohne Angabe von Gründen besser gefiel (SW0806) und das deren Steuerung / Bedienung besser fand (SW0807), empfand die optische Aufbereitung der Software Focus on Bee-Bot 1 als ansprechender (SW0808). Ein Kind, dem die Software Focus on Bee-Bot 2 ohne Angaben von Gründen besser gefiel (SW0806) und das keine Angaben machte welche Steuerung / Bedienung es bevorzugte (SW0807) empfand die optische Aufbereitung der Software Focus on Bee-Bot 2 als ansprechender (SW0808).

9.2.6.8.2 SWT2

2 der 8 an der Usability Evaluierung SWT2 teilnehmenden Kinder erkannten das Vorhandensein zweier GO-Buttons in der Software Focus on Bee-Bot 2, 5 Kinder erkannten dies nicht (SW0801), ein Kind kam nicht in die Situation, die Software Focus on Bee-Bot 2 zu testen, da es unerwartet vorzeitig von seinen Eltern abgeholt wurde, was sich bezüglich der Überkategorie „Unterschiede“ neben der Kodierung der Kategorie SW0801 auch auf die Kodierung der Kategorien SW0802, SW0803, SW0804, SW0805, SW0806, SW0807 und SW0808 auswirkte.

Ein Kind thematisierte die Unterschiede der Links- und Rechts-Dreh-Buttons der beiden Software-Versionen (SW0805). Bezüglich der Unterschiede zwischen den Software-Versionen wurde den Kindern die offene Frage gestellt, ob ihnen eine Software besser gefallen hatte und wenn ja aus welchem Grund / aus welchen Gründen. Aus den Nennungen der Kinder ergeben sich für den SWT2 folgende Ergebnisse: ein Kind fand die Matten in der Software Focus on Bee-Bot 2 schöner / besser (SW0802), zur ansprechenderen optischen Aufbereitung einer Software Version äußerte sich kein Kind (SW0808). Das Kind, das die Matten in der Focus on Bee-Bot 2 Software präferierte (SW0802) empfand gemeinsam mit einem anderen Kind die Software Focus on Bee-Bot 1 als leichter zu bedienen (SW0803), 2 weitere Kinder empfanden dies für die Software Focus on Bee-Bot 2 (SW0803), wovon ein Kind auch angab, dass der Navigationsbereich der Software Focus on Bee-Bot 2 leichter zu bedienen sei (SW0804). 6 Kindern gefiel im SWT2 die Software Focus on Bee-Bot 2 ohne Angaben von Gründen besser, eines dieser Kinder präferierte die Steuerung / Bedienung dieser Software Version (SW0807). Einem Kind gefiel „alles“, es fand beide Software Versionen ohne Angaben von Gründen gut (SW0806).

9.2.6.9 SW09 Wissenstransfer

9.2.6.9.1 SWT1

4 der 9 an der Usability Evaluierung SWT1 teilnehmenden Kinder erkannten, dass die in der Software verwendeten Matten grundsätzlich so aussehen, wie die in Natura zur Verfügung stehenden (SW0901). Diese 4 Kinder und 3 weitere thematisierten Unterschiede zum realen

Bee-Bot (SW0902). Von diesen 7 Kindern thematisierten 5 Kinder das Zurückstellen des Bee-Bots in der Software auf das Startfeld, wenn der GO-Button gedrückt wurde und der Bee-Bot wo anders gestanden hat, als auf dem Startfeld als Unterschied zum realen Bee-Bot (SW0912). Kein Kind thematisierte, dass der reale Bee-Bot stechen kann oder böse ist (SW0903).

Alle 9 am SWT1 partizipierenden Kinder erkannten den Bee-Bot in der Software als Bee-Bot wieder (SW0904). 4 Kinder erkannten Unterschiede bei den Buttons zwischen dem realen Bee-Bot und dem Navigationsfeld der Software Focus on Bee-Bot 1 (SW0905). 2 dieser Kinder gaben auch Unterschiede zwischen den Buttons des realen Bee-Bots und den Buttons des Navigationsfelds der Software Focus on Bee-Bot 2 an (SW0905). Neben diesen 2 Kindern und einem Kind, das Unterschiede zwischen dem realen Bee-Bot und der Software Focus on Bee-Bot 1 nannte (SW0905), erkannten weitere 4 Kinder, die keine Unterschiede zwischen den Tasten des realen Bee-Bots und den Buttons des jeweiligen Navigationsfelds der beiden Software Versionen anführten (SW0905), Unterschiede bei den Buttons zwischen den beiden Software Versionen (SW0906). Kein Kind gab an, durch die jeweils aufgefundenen Unterschiede irritiert gewesen zu sein (SW0907).

7 Kinder konnten im SWT1 die Erfahrung / das Wissen aus der Nutzung der Focus on Bee-Bot 1 Software in die Nutzung der Focus on Bee-Bot 2 Software übertragen (SW0908). 5 dieser 7 Kinder konnten sowohl die Erfahrung / das Wissen aus der Nutzung der Hardware (Bee-Bot) in die Nutzung der Software übertragen (SW0910), als auch die Erfahrung / das Wissen aus der Nutzung des Bee-Bots und der Software Focus on Bee-Bot 1 bzw. 2 auf den Constructa-Bot übertragen (SW0911), ein weiteres dieser 7 Kinder konnte die Erfahrung / das Wissen aus der Nutzung der Hardware (Bee-Bot) in die Nutzung der Software übertragen (SW0910). Noch eines dieser 7 Kinder konnte die Erfahrung / das Wissen aus der Nutzung des Bee-Bots und der Software Focus on Bee-Bot 1 bzw. 2 auf den Constructa-Bot übertragen (SW0911). 2 Kinder, die das Wissen aus der einen Software Version nicht in die andere Version übertragen konnten (SW0908), konnten die Erfahrung / das Wissen aus der Nutzung der Hardware (Bee-Bot) in die Nutzung der Software übertragen (SW0910), eines dieser Kinder konnte auch die Erfahrung / das Wissen aus der Nutzung des Bee-Bots und der Software Focus on Bee-Bot 1 bzw. 2 auf den Constructa-Bot übertragen (SW0911).

Ein Kind thematisierte im SWT1 bei der Focus on Bee-Bot Software, dass Buttons ausgegraut waren (im Moment nicht verfügbar) als Unterschied zum realen Bee-Bot (SW0913). 4 Kindern wurde zur Lösung einer gestellten Aufgabe mit einer Focus on Bee-Bot Software Version die Lösung (etwa bezüglich Richtungsentscheidungen) von TL in der realen Welt veranschaulicht, 2 von ihnen fiel es schwer, die Lösung der Aufgabe aus der realen Welt in die virtuelle Welt zu übertragen (SW0909).

9.2.6.9.2 SWT2

Die Kategorien SW0901, SW0902, SW0904, SW0905, SW0910, SW0911, SW0912 und SW0913 wurden aufgrund der (absichtlich) fehlenden Vorerfahrung mit dem realen Bee-Bot der an der Usability Evaluierung teilnehmenden Kinder im SWT2 nicht kodiert. 3 der 8 an der Usability Evaluierung im SWT2 involvierten Kinder thematisierten, dass der reale Bee-Bot stechen kann bzw. böse ist (SW0903). 3 Kinder erkannten Unterschiede bei den Buttons zwischen den beiden Software-Versionen (SW0906), eines dieser Kinder gab an, durch die jeweils aufgefundenen Unterschiede irritiert gewesen zu sein (SW0907). 2 dieser 3 Kinder und 4 weitere konnten die Erfahrung / das Wissen aus der Nutzung der Focus on Bee-Bot 1 Software in die Nutzung der Focus on Bee-Bot 2 Software übertragen (SW0908).

4 Kindern wurde im SWT2 zur Lösung einer gestellten Aufgabe mit einer Focus on Bee-Bot Software Version die Lösung (etwa bezüglich Richtungsentscheidungen) von TL in der realen Welt veranschaulicht, 2 von ihnen fiel es schwer, die Lösung der Aufgabe aus der realen Welt in die virtuelle Welt zu übertragen (SW0909).

9.2.6.10 SW10 Präferenzen

9.2.6.10.1 SWT1

Alle 9 an der Usability Evaluierung SWT1 teilnehmenden Kinder empfanden die Funktion „Einzelschritte anzeigen“ als hilfreich (SW1001). 8 Kindern gefiel der Constructa-Bot besser als der Bee-Bot (SW1003), ein Kind empfand den Bee-Bot ansprechender als den Constructa-Bot (SW1002). Dieses Kind und ein weiteres bevorzugten nicht die Software gegenüber dem realen Bee-Bot, was 5 Kinder im SWT1 taten (SW1004). 2 Kinder gaben hierzu keine Präferenzen an (SW1004).

9.2.6.10.2 SWT2

Die Fragen zu den Kategorien SW1002 und SW1003 wurden aufgrund der (absichtlich) fehlenden Vorerfahrung mit dem realen Bee-Bot der in der Usability Evaluierung SWT2 involvierten Kinder nicht gestellt.

5 der an der Usability Evaluierung teilnehmenden Kinder empfanden die Funktion „Einzelschritte anzeigen“ als hilfreich (SW1001). Befragt nach der Präferenz gegenüber dem realen Bee-Bot (den die teilnehmenden Kinder noch nicht gesehen hatten), bevorzugten 2 Kinder die Software gegenüber dem realen Bee-Bot (SW1004), darunter ein Kind, das thematisierte, dass der reale Bee-Bot stechen kann bzw. böse ist (SW0903). Die 2 weiteren Kinder, die thematisierten, dass der reale Bee-Bot stechen kann bzw. böse ist (SW0903), bevorzugten gemeinsam mit einem anderen Kind nicht die Software gegenüber dem realen Bee-Bot (SW1004), die weiteren Kinder machten diesbezüglich keine Angaben.

9.2.6.11 SW11 Testsetting

9.2.6.11.1 SWT1

Alle 9 an der Usability Evaluierung SWT1 partizipierenden Kinder interagierten mit TL (SW1104), wovon 8 ihr eigenes Handeln kommentierten (SW1102). Bis auf ein Kind stellten alle Kinder von sich aus Fragen an TL (SW1103), dieses Kind und 2 weitere ließen sich während einer Aufgabe durch andere Programmelemente ablenken (SW1106). Eines dieser 2 weiteren Kinder ignorierte während der Evaluierung einmal oder mehrmals die Aufgabenstellung und klickte auf der Oberfläche der Focus on Bee-Bot Software (SW1105). Ein Kind meinte während des SWT1 eine gestellte Aufgabe nicht lösen zu können (SW1107).

9.2.6.11.2 SWT2

Im SWT2 stellten sich 3 der 8 teilnehmenden Kinder selbst (weitere) Aufgabenstellungen mit dem Bee-Bot in der Software (SW1101). 7 Kinder kommentierten ihr eigenes Handeln (SW1102), 6 dieser 7 Kinder interagierten mit TL (SW1104). Alle in die Usability Evaluierung SWT2 involvierten Kinder stellten von sich aus Fragen an TL (SW1103). 2 Kinder, die nicht mit TL interagierten (SW1104) und 3 weitere ließen sich während einer Aufgabe durch andere Programmelemente ablenken (SW1106). 3 dieser 5 Kinder, davon 2 die nicht mit TL interagierten (SW1104), ignorierten während der Evaluierung einmal oder mehrmals die Aufgabenstellung und klickten auf der Oberfläche der Focus on Bee-Bot Software (SW1105). 3 Kinder meinten im SWT2, dass sie eine gestellte Aufgabe nicht lösen könnten (SW1107).

9.2.7 Zusammenfassende Interpretation der Ergebnisse und Schlussfolgerungen

In Bezug auf das Vorwissen (SW01) sind die beiden Testgruppen des SWT1 und SWT2 differenziert zu betrachten. Beim SWT1 gaben beinahe alle Kinder an, dass sie bereits im Vorfeld einmal einen Computer genutzt oder mit einem Computer gespielt haben (SW0101). Keines der Kinder kannte jedoch nach eigenen Angaben eine Software, die ähnlich zu der im Testsetting war (Focus on Bee-Bot 1 bzw. 2). Rund die Hälfte der Kinder beim SWT1 kannte bereits eine Computermouse (SW0105) und ebenfalls rund die Hälfte der Kinder konnte mit

der Computermaus soweit umgehen, dass ihnen das Auswählen von Elementen am Bildschirm mit der Computermaus keine Probleme bereitete. Nur rund die Hälfte der Kinder, die am SWT2 teilnahmen, gab zu Beginn der Tests an, dass sie schon einmal einen Computer genutzt bzw. mit einem Computer gespielt haben. Ein Kind meinte, dass es schon einmal mit einer ähnlichen Software wie der Focus on Bee-Bot 1 bzw. 2 gespielt hatte. Ebenfalls rund die Hälfte der Kinder konnte die Computermaus bestimmungsgemäß nutzen.

Es zeigt sich somit, dass durchaus entsprechendes Vorwissen bei den Kindern in der Altersgruppe der 3- bis 6-Jährigen vorhanden ist und auf dieses Vorwissen, zumindest auf individueller Basis pro Kind, gut aufgebaut werden kann.

Die Wahrnehmung der zentralen Figur (SW02), der Biene, in der Software Focus on Bee-Bot 1 bzw. 2 durch die Kinder ist ebenfalls nach Testgruppe divers zu betrachten. Die Kinder der Gruppe, die am SWT1 teilnahmen, wussten bereits vom vorangegangenen Hardware-Test, dass es sich bei der zentralen Figur um eine Biene handelt (SW0201). Sie wussten auch, dass diese fahren bzw. sich auf einer ebenen Fläche bewegen kann (SW0205). Dennoch meinte knapp die Hälfte der Kinder, dass diese Biene in der Software springen kann (SW0206) und rund ein Drittel meinte, dass sie auch fliegen kann (SW0207). In der Gruppe, die am SWT2 teilnahm, erkannte nur rund ein Drittel der Kinder die zentrale Figur als Biene. Die Hälfte der Kinder thematisierte, dass der abgebildete Bee-Bot, als zentrale Figur, ein Roboter ist (SW0202). Ein Kind war überzeugt, dass es sich um ein Tier und keinen Roboter handelt, ob der Tatsache, dass die zentrale Figur in Form eines Tiers erscheint (SW0203, SW0204). Rund die Hälfte der Kinder interpretierte, dass der Bee-Bot am Bildschirm fahren kann. Aber auch die Möglichkeiten, dass der Bee-Bot springen oder fliegen kann, wurden von rund einem Drittel der Kinder genannt.

Die Mehrheit der Kinder erkannte im Bee-Bot als zentrale Figur am Bildschirm eine Biene bzw. ein Tier. Naturgemäß taten sich hierbei die Kinder leichter, die den Bee-Bot schon aus dem Hardware-Test kannten.

Je nach Testgruppe erkannte gerade einmal ein Drittel der Kinder, dass im linken Bildschirmbereich eine Befehlsliste vorhanden ist (SW0209), die den Speicher des Bee-Bots simuliert. Alle Kinder fokussierten ihre Aufmerksamkeit zu Beginn in die Mitte des Bildschirms (SW0212) und nahmen verstärkt die dort befindlichen Elemente wahr. Die vom Zentrum weiter entfernten Elemente wurden nur nach und nach erkundet, oft erst nach entsprechenden Fragestellungen und Aufforderungen durch die Testleitung.

Beinahe alle Kinder erkannten, dass die Software Focus on Bee-Bot 1 bzw. 2 mit der Computermaus zu bedienen ist (SW0301). Während der Bedienung und der Usability-Tests vermutete rund die Hälfte der Kinder, dass es noch andere Eingabegeräte als die Computermaus geben könnte und versuchten via Tastatur, TouchScreen oder ähnlichem den Bee-Bot zu steuern (SW0302). Mehr als die Hälfte der Kinder wollte den Bee-Bot über die auf seinem Rücken aufgezeichneten Buttons bedienen (SW0308, SW0311). Auch der Versuch den Bee-Bot über Klick auf die dargestellte Matte zu bewegen wurde von einigen Kindern unternommen (SW0312), ebenso wie die Nutzung von Drag-&Drop-Aktionen oder schieben mit der Computermaus (SW0318). Die Bedienung des Bee-Bots am Bildschirm stellt somit eine Herausforderung für die Kinder dar. Zum einen erscheint ihnen die alleinige Nutzung der Computermaus zur Bedienung zu wenig, andererseits suggeriert das Programm durch die Gestaltung der zentralen Figur mit ihren aufgemalten Tasten am Rücken eine weitere Bedienungsmöglichkeit als die ursprünglich vorgesehene Eingabe von Richtungsbefehlen via Mausclicks auf abgebildete Richtungs-Buttons im Eingabebereich links oben.

Sowohl die Bedeutung als auch das Auffinden der Buttons GO und Clear stellte für die Kinder eine Hürde in der Bedienung dar. Die Bedeutung der Buttons konnte erst durch Erklärung durch die Testleitung vermittelt werden. Die Position selbiger ebenso. Positiv bewerteten die Kinder die Tatsache, dass der Bee-Bot bei Betätigung des GO-Buttons

wieder von seinem ursprünglichen Startpunkt aus die Abarbeitung der eingespeicherten Befehle vornahm (SW0322).

Die im Verlauf des Tests aufgetretenen (Text-)Pop-Ups konnte nur ein Kind lesen bzw. sinngemäß interpretieren (SW0323) – hier ist deutlich auf die Möglichkeit einer grafischen Gestaltung hinzuweisen. Mehr als die Hälfte der Kinder hatten Probleme bei der Interpretation der Links- und Rechts-Dreh-Buttons zur Steuerung des Bee-Bots. Sie glaubten, dass es sich hierbei um einen Schritt nach links bzw. rechts handle und nicht um eine Drehung um 90 Grad nach links bzw. rechts (SW0305). Hier wäre eine geeignetere grafische Darstellung zu wählen. Wiederum die Hälfte der Kinder konnte den Button für „Einzelschritte anzeigen“ mit maximal einem Hinweis auffinden (SW0316), was für eine geeignete Darstellung zur sinngemäßen Interpretation spricht.

Es lassen sich nur schwer verallgemeinerbare Aussagen treffen – zu sehr sind die Einzelergebnisse von Vorerfahrungen der Kinder abhängig. Es zeigt sich jedoch schon nach den ersten Testabschnitten, dass es für Kinder schwer ist, die Software ohne entsprechende Anleitung bzw. Instruktionen zu nutzen, was mit den Ergebnissen der Kategorie SW04 weiter belegt wird.

Die Kinder, die zuvor noch an keinem Hardware-Usability-Test des Bee-Bots teilgenommen haben oder den Bee-Bot selbst noch nicht kennengelernt hatten (SWT2), konnten die Software nicht selbständig ohne Hilfestellungen sinngemäß nutzen (SW0401, SW0402, SW0403). Die Kinder, die den Hardware-Bee-Bot kannten, konnten nach einer kurzen Umgewöhnungsphase rasch erste Ergebnisse erzielen (SWT1). Auch das Lösen von Aufgaben und die Nutzung weiterer Buttons, die nicht zur Richtungssteuerung genutzt werden, fiel den Kindern aus der Testgruppe des SWT1 wesentlich leichter.

Die Kenntnis und korrekte Interpretation der Funktionalität von Löschen bzw. „vergessen“ war, nachvollziehbarer Weise, bei der Gruppe, die am SWT1 teilnahm deutlich stärker ausgeprägt, als bei der Testgruppe des SWT2 – wobei hier rund die Hälfte der Kinder diese korrekt interpretierte und nutzen konnte (SW0409, SW0411). Kaum ein Kind erkannte die Unterschiede zwischen aktiven und nicht-aktiven Buttons (SW0410). Die Funktionalität „Einzelschritte anzeigen“ bzw. „Einzelschritte aus“ erschloss sich den Kindern nur nach Hinweisen und Instruktionen.

Die Kinder des SWT1 zeigten bei der Orientierung (auf der Matte am Bildschirm und in Korrelation der zentralen Figur, dem Bee-Bot, zu der Matte und den Bedienelementen SW05) eine höhere Leistung als die Kinder des SWT2. Die Kinder des SWT1 waren, natürlich auf Grund der vorhergegangenen Beschäftigung mit dem Hardware-Bee-Bot, eher in der Lage, den Bee-Bot zu navigieren und die unterschiedlichen Positionen des Bee-Bots in Korrelation zu seiner Umgebung korrekt zu interpretieren. Große Probleme hatten die Kinder, die am SWT2 teilnahmen, mit der Orientierung, wenn der Bee-Bot in Relation zum Navigationsbereich nicht in die gleiche Richtung blickte (SW0502). Rund ein Achtel der Kinder konnte sich bei der Orientierung durch eigenständige Rotation der Matte behelfen, so dass die Interpretation der Richtungen des Bee-Bots vereinfacht wurde (SW0501).

Hier zeigt sich, dass es anfangs für die Kinder deutlich schwieriger ist, sich im virtuellen Raum zu orientieren, als wenn sie den Bee-Bot angreifen und die „Orientierung“ bzw. „Richtungsbewegung“ erleben können. Diese Erlebnisse helfen ihnen, sich anschließend im virtuellen Raum korrekt zu bewegen und zu orientieren.

Im Verlauf des Software Usability Tests zeigte sich, dass den Kindern die Mausbedienung der Software schwerfiel (SW06). Herausfordernd war für sie vor allem die Positionierung des Cursors während eines Mausklicks. Wenn die Kinder einen Mausklick ausüben wollten, führten ihre Motorik und vermutlich auch die kleinen Hände oftmals dazu, dass die Bewegung des Fingers beim Klicken die Computermaus bewegte und diese somit auch vom Ziel des Klicks am Bildschirm wegrutschte.

Öfters kam es vor, dass Kinder, zum großen Teil unabsichtlich, die Matte, auf der sich der virtuelle Bee-Bot bewegt, mit der Maus drehten (SW07, SW0701). Ein gutes Drittel der Kinder, bei denen es zu einer Drehung der Matte kam, konnte entweder die Drehung stoppen oder die Matte wieder in die Ausgangsposition drehen, wobei manche Kinder beides konnten (SW0702, SW0704).

Bezüglich der erkannten Unterschiede zwischen den beiden Software-Versionen und der Vorlieben der Kinder für eine der beiden Versionen (SW08, SW09, SW10) sind keine verallgemeinerbaren Aussagen zu treffen. Die Vorlieben sind hier sehr individuell und oftmals auch ohne Grund genannt worden. Nur rund einem Drittel der Kinder fiel auf, dass es in der Software Focus on Bee Bot 2 zwei GO-Buttons gab, gegenüber nur einem in der Software Focus on Bee Bot 1 (SW0801). Auch sonst wurden Unterschiede nur selten erkannt.

Die Reaktion und die Interaktion der Kinder im Testsetting (SW11) sind sehr positiv zu beurteilen. Die Kinder waren interessiert und beteiligten sich überaus aktiv am Testgeschehen. Vereinzelt wurden Aufgabenstellungen ignoriert oder war ein Kind komplett überfordert (was z.B. bei SWT2 zu einem Abbruch der Kodierung führte, da die Ergebnisse nicht verwertbar waren, siehe auch Kapitel 9.2.6). Das Testsetting erscheint somit für die Altersgruppe geeignet.

Weitere Detailergebnisse im direkten Vergleich der beiden Testgruppen liefert der folgende Abschnitt.

9.2.7.1 Exemplarische Unterschiede zwischen SWT1 und SWT2

Die einzelnen Unterkategorien wurden Kategorie für Kategorie in Hinblick auf die beiden Gruppen, die am SWT1 und SWT2 teilnahmen, verglichen und im Anschluss finden sich exemplarische Unterschiede in den Ergebnissen.

Alle Kinder des SWT1 und 37,5 % der Kinder des SWT2 erkannten im Bee-Bot eine Biene (SW0201). Rund 11 % der Kinder des SWT1 erkannten im Bee-Bot ein anderes Tier, dies war auch bei 62,5 % der Kinder des SWT2 der Fall (SW0204). Alle Kinder des SWT1 interpretierten, dass der Bee-Bot im Programm fahren kann, dies deuteten rund 57,1 % der Kinder des SWT2 (SW0205).

Rund 67 % der Kinder des SWT1 interpretierten, dass das Programm mit anderen Eingabegeräten (Tastatur, TouchPad, TouchScreen, Lenkrad, Fernsteuerung) zu bedienen ist, dies deuteten 25 % der Kinder des SWT2 (SW0302). Rund 11 % der Kinder des SWT1 versuchten, das Programm mit Spracheingabe zu bedienen, diesen Versuch unternahm 62,5 % der Kinder des SWT2 (SW0303). Rund 22 % der Kinder des SWT1 hatten keine Probleme den Clear- und Pause-Button der Focus on Bee-Bot Software 2 zu unterscheiden. Im SWT2 hatten rund 83,3 % der Kinder, für die diese Kategorie kodiert werden konnte, diesbezüglich keine Probleme (SW0304). Rund 56 % der Kinder des SWT1 fassten den Links- und Rechts-Dreh-Button als 90 Grad Drehung und nicht als Schritt nach links bzw. rechts auf. Dies deuteten rund 14,3 % der Kinder des SWT2, für die diese Kategorie kodiert werden konnte (SW0305). Rund 89 % der Kinder des SWT1 und rund 42,9 % der Kinder des SWT2 interpretierten den GO-Button als „Start“ für das Losfahren des Bee-Bots im Programm (SW0306). Dass der Bee-Bot im Navigationsbereich auf dem Rücken bedient werden kann, deuteten rund 78 % der Kinder des SWT1, dies nahmen 37,5 % der Kinder des SWT2 an (SW0311). Während rund 78 % der Kinder des SWT1 den Button für Clear auf Anhieb mit max. einem Hinweis fanden, war dies bei 37,5 % der Kinder des SWT2 der Fall (SW0314).

Rund 67 % der Kinder des SWT1 interpretierten die Funktionalität von GO sinngemäß als Abarbeiten der gespeicherten Befehle, bis der Speicher mit Clear gelöscht wird, dies deuteten 12,5 % der Kinder des SWT2 (SW0401). Bei der Software Focus on Bee-Bot 2 wurde der GO-Button im Navigationsbereich von rund 78 % der Kinder des SWT1 häufiger genutzt als der in der Symbolleiste. Dies war bei rund 28,6 % der Kinder des SWT2, die zu

dieser Kategorie kodiert werden konnten, der Fall (SW0402). Rund 56 % der Kinder des SWT1 speicherten eine für die Aufgabenstellung geeignete Abfolge von Richtungsbefehlen ohne Hilfestellung ein, dies erfolgte bei keinem Kind im SWT2 (SW0403). Das Hindernis in der Software Focus on Bee-Bot 1 erkannten rund 71,4 % der Kinder des SWT1, für die diese Kategorie kodiert werden konnte, als unüberwindbares Hindernis und umfuhren es, was bei 12,5 % der Kinder des SWT2 der Fall war (SW0408). Rund 85,7 % der Kinder des SWT1, für die diese Kategorie kodiert werden konnte, simulierten „Einzelschritte anzeigen“ durch Betätigen des GO-Buttons nach (fast) jeder (Richtungs-)Befehls-Eingabe, so gingen rund 28,6 % der Kinder des SWT2, für die diese Kategorie kodiert werden konnte, vor (SW0413).

Rund 44 % der Kinder des SWT1 bedienten die Richtungs-Buttons trotz anderer Blickrichtung des Bee-Bots in Relation zum Navigationsbereich korrekt, dies war bei keinem Kind des SWT2 der Fall (SW0502). Es schien bei allen Kindern des SWT1, dass ihnen die Bedienung des Programms leichter fiel, wenn der Bee-Bot in Relation zum Navigationsfeld in die gleiche Richtung sah, dies konnte bei rund 57,1 % der Kinder des SWT2, für die diese Kategorie kodiert werden konnte, beobachtet werden (SW0503). Rund 78 % der Kinder des SWT1 konnten Richtungsfehler selbst erkennen und korrigieren, 37,5 % der Kinder konnten dies beim SWT2 (SW0505).

Beim SWT1 kamen rund 67 % der Kinder näher zum Bildschirm, um den Inhalt besser sehen zu können, beim SWT2 war dies bei 12,5 % der Kinder der Fall (SW0603).

Rund 56 % der Kinder des SWT1 und 12,5 % des SWT2 machten Angaben darüber, welche Matten welcher Software sie bevorzugten (SW0802). Angaben darüber, welchen Navigationsbereich sie als leichter empfanden, machten im SWT1 rund 56 % der Kinder, im SWT2 12,5 % der Kinder (SW0804). Präferenzen bezüglich der Steuerung / Bedienung gaben rund 78 % der Kinder des SWT1 und 12,5 % des SWT2 an (SW0807).

Wie bereits in Kapitel 9.2.6.9 dargelegt, zeigte sich, dass die Kinder des SWT1 bei der Bedienung des Bee-Bots auf ihr zuvor angeeignetes Wissen zurückgreifen konnten und ihnen diese daher in Folge auch leichter fiel (SW09). Dennoch lösten Kinder aus der Testgruppe, die am SWT2 teilnahmen mit Anleitung und Hilfestellungen erste Aufgaben. Auch wurden von Kindern aus der Testgruppe des SWT1 mehr Angaben zu Präferenzen gemacht, als von Kindern aus der Testgruppe des SWT2, vermutlich, da sie mit einem Testsetting ähnlicher Art schon vertraut waren.

9.2.7.2 Exemplarische geschlechterspezifische Unterschiede (unabhängig von SWT1 und SWT2)

Die einzelnen Unterkategorien wurden Kategorie für Kategorie in Hinblick auf die beiden Geschlechter verglichen, im Anschluss finden sich exemplarische Unterschiede in den Ergebnissen.

60 % der Jungen interpretierten, dass der Bee-Bot in der Software fliegen kann, dies deutete keines der Mädchen (SW0207).

80 % der Jungen und rund 30% der Mädchen vermuteten, dass der Bee-Bot im Navigationsbereich auf dem Rücken bedient werden kann (SW0311). 10 % der Jungen und rund 57 % der Mädchen interpretierten, dass der Bee-Bot durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung gesteuert werden kann (SW0312). 10 % der Jungen fanden den Button für „Einzelschritte anzeigen“ auf Anhieb mit max. einem Hinweis, diesen fanden rund 71 % der Mädchen auf Anhieb mit max. einem Hinweis (SW0316). 50 % der Jungen interpretierten einen beliebigen Button (außer „Einzelschritte anzeigen“ selbst) als Button für „Einzelschritte anzeigen“, dies deutete kein Mädchen (SW0317).

30 % der Jungen machten Angaben darüber, welche Focus on Bee-Bot Software sie leichter (zur Bedienung) fanden, bei den Mädchen war dies bei rund 86 % der Fall (SW0803). Rund 57 % der Mädchen empfanden den Navigationsbereich der Focus on Bee-Bot 2 Software leichter (in der Bedienung), dies empfand kein Junge (SW0804).

Kein Mädchen und 40 % der Jungen ignorierten eine Aufgabenstellung oder mehrere und klickten auf der Oberfläche der Software (SW1105). 60 % der Jungen ließen sich durch andere Programmelemente während einer Aufgabe ablenken, dies war bei rund 14 % der Mädchen der Fall (SW1106).

Aus dem direkten Vergleich lässt sich ableiten, dass Jungen eher experimentierfreudig sind, weniger der Aufgabenstellung folgen und sich sehr leicht durch andere Elemente am Bildschirm ablenken lassen – also etwas aufgeregter und empfänglicher erscheinen. Mädchen hingegen wirken etwas besonnener, geduldiger und lassen sich bei der Lösung einer Aufgabe weniger leicht ablenken. Sie hören auch eher der Aufgabenstellung zu. Sie sind darüber hinaus kommunikativer, wenn es um die Mitteilung von Präferenzen geht.

9.2.7.3 Beitrag der Software Usability Evaluierung zur frühen Technikbildung in Informatik

Die Kinder werden durch ihre Teilnahme an der Evaluierung der Software Usability in unterschiedlichen Bereichen (korrespondierend zu den Lernergebnissen aus Kapitel 8.2) gefördert. Einer dieser Bereiche ist das PC-Anwenderwissen. Die Kinder üben Objekte am Bildschirm mit dem Cursor via Eingabegerät anzuvisieren und zu selektieren, einen Programmbefehl mittels einfachem Klick aufzurufen, ein Programm mittels Doppelklick zu öffnen, einen Befehl in der Menüleiste auszuwählen und aufzurufen, bei auftretenden Pop-Up-Fenstern die angezeigten Schaltflächen zu bedienen und einen Zusammenhang zwischen den Aktionen, die sie mittels Eingabegerät „einspeichern“ und dem was im Programm passiert herzustellen (Ursache-Wirkung) (vgl. LE2.4).

Weiter trainieren die Kinder ihr eigenes Handeln zu kommentieren (vgl. LE3.5), Programme hinsichtlich ihrer Bedienung zu vergleichen, zu beschreiben was sie sehen, durch „trial and error“ die Bedienung eines ihnen unbekanntes Programms zu erkunden, offene Fragen und Aufgabenstellungen zu beantworten bzw. zu bearbeiten (vgl. LE3.4) sowie sich über einen Zeitraum von 20 Minuten einer Aufgabe zu widmen.

Zusätzlich ergeben sich für die Kinder die Möglichkeiten einfache und komplexe Programmieraufgaben zu lösen (vgl. LE2.1), Parallelen und Unterschiede zwischen der Bedienung von Hardware und korrespondierender Software zu identifizieren und zu benennen, an themenrelevantes Vorwissen anzuknüpfen (vgl. LE1.7) und dieses abzurufen, sowie unterschiedliche Perspektiven bei der Programmierung der Simulation des Bodenroboters in der Mikrowelt einzunehmen (vgl. LE2.12).

Durch die aktive Mitarbeit an der Evaluierung der Software Usability werden die Kinder motiviert, die grundlegende Funktionalität der Simulation des Bodenroboters in dessen Mikrowelt zu erklären (vgl. LE2.9), die Simulation eines Bodenroboters zur Lösung von Problemstellungen in dessen Mikrowelt zu bedienen (vgl. LE2.10), mit der Testleitung zu interagieren sowie informatische und technische Begriffe in eigenen Worten zu formulieren bzw. wiederzugeben (vgl. LE1.1 und LE2.2).

10 Child-Centered-Design im Kindergarten

Zur empirischen Befundung, ob ein adaptierter Child-Centered-Design-Ansatz, der auf den bisher aufgefundenen Ergebnissen der gegenständlichen Forschungsarbeit basiert, zur frühen Technikbildung in Informatik im Kindergarten eingesetzt werden kann, führt die Autorin selbigen am Beispiel der Entwicklung und Evaluation eines Prototyps einer grafischen Benutzeroberfläche einer Mikrowelt (siehe Kapitel 4.3.1.1), die als Lernumgebung zum Erlernen der Programmierung für 3- bis 6-jährige Kinder geeignet ist, durch. Bei der empirischen Befundung steht vor allem der Child-Centered-Design-Prozess selbst im Vordergrund des Interesses.

Hauptziel ist, dass die am adaptierten Child-Centered-Design-Ansatz teilnehmenden Kinder als Lernende die unterschiedlichen Facetten der Software Entwicklung kennenlernen und hierbei unterschiedliche Rollen im Design-Prozess einnehmen. Zusätzlich entsteht ein Prototyp einer grafischen Benutzeroberfläche einer Mikrowelt, die als Lernumgebung zum Erlernen der Programmierung für 3- bis 6-jährige Kinder geeignet ist. Darüber hinaus haben die Kinder die Möglichkeit die Tätigkeit einer Forscherin / eines Forschers durch die Aktivitäten der Autorin, welche diese den Kindern kindgerecht erklärt, zu beobachten.

Als Ausgangspunkt dient das klassische User-Centered-Design (siehe Kapitel 5.2.2), wobei durch die Ausrichtung an der Zielgruppe der 3- bis 6-jährigen Kinder auch von Child-Centered-Design gesprochen werden kann.

Dieses klassische Vorgehen, das nur die Ausrichtung der Usability-Aktivitäten an den Benutzerinnen und Benutzern vorsieht (siehe Kapitel 5.2.2), dient als Basis für die Kombination von Cooperative-Inquiry und Bonded-Design (siehe Kapitel 5.2.3.3) zum gegenständlichen adaptierten Child-Centered-Design, ergänzt um Inspirationen aus Best Practice Berichten (siehe Kapitel 5.4 und 6.3). Hierbei werden die Benutzerinnen und Benutzer (Kindergartenkinder) schon viel früher als aktive Partnerinnen und Partner im Prozess integriert und als vollwertige Gestalterinnen und Gestalter angesehen.

Als Innovation werden die Kinder vorrangig als Lernende der Informatik im Speziellen der Software Entwicklung betrachtet. Zum Lernen nehmen die Kinder in diesem adaptierten Child-Centered-Design unterschiedliche Rollen (siehe Kapitel 5.2.1) ein. Zum einen sind die Kinder Benutzerinnen und Benutzer, die Zielgruppe des als Lernobjekt erstellten Prototyps. Zum anderen agieren sie als Testerinnen und Tester dieses Prototyps, denn mit Hilfe ihrer Rückmeldungen soll als Lernsituation eine Iteration zur Verbesserung des Prototyps durchgeführt werden. Die Kinder sind auch als Informantinnen und Informanten beteiligt, da die Autorin den Kindern zeigen möchte, wie durch den Einsatz von Ermittlungsmethoden (im Kontext von Kindergärten verpackt in Spiele bzw. eingebettet in Spiele) Informationen über Vorlieben und Ideen sowie Wünsche von den Benutzerinnen und Benutzern gewonnen werden können. In der vierten Rolle, als Gestalterinnen und Gestalter, sind die Kinder bei der Gestaltung von Symbolen für die grafische Benutzeroberfläche des exemplarisch erstellten Prototyps eingebunden.

Mit Kapitel 10.1 beginnt die Autorin den adaptierten Child-Centered-Design-Prozess als Maßnahme zur frühen Technikbildung in Informatik vorzustellen. Sie präsentiert dessen Ablauf, durch den die teilnehmenden Kinder als Lernende die unterschiedlichen Bereiche der Software Entwicklung erleben. Die ersten Artefakte, die die Kinder erstellen, sind Symbole, die im Lernobjekt Prototyp zum Einsatz kommen sollen (Kapitel 10.2). Hier agieren die Kinder in der Rolle der Gestalterinnen und Gestalter, sowie auch als Informantinnen und Informanten. Anschließend führt die Verfasserin der gegenständlichen Arbeit ein Card-Sorting Verfahren durch, um den Kindern zu zeigen, wie Symbole ihren jeweiligen Bedeutungen korrekt zugeordnet werden können (Kapitel 10.3). Dabei agieren die Kinder in der Rolle der Testerinnen und Tester, sowie als Informantinnen und Informanten. Die Entwicklung des ersten Prototyps, seine Evaluierung und seine Evolution in Version 2, gefolgt von einer weiteren Evaluierung beschreibt die Autorin in Kapitel 10.4 – hier agieren die Kinder als Benutzerinnen und Benutzer sowie als Testerinnen und Tester. Der Beitrag,

den Child-Centered-Design zur frühen Technikbildung in Informatik für 3- bis 6-jährige Kinder leisten kann, wird von der Verfasserin der gegenständlichen Arbeit in Kapitel 10.5 beleuchtet.

10.1 Ablauf des Child-Centered-Designs

Basierend auf den Erfahrungen aus den Lehr-Lern-Einheiten in den Kindergärten D und E und den dort durchgeführten Aktivitäten, sowie auf den Berichten aus der Literatur (siehe Kapitel 5.2 und 5.3), entschied die Autorin, den adaptierten Child-Centered-Design-Prozess als Technikbildungsmaßnahme in der Informatik mit 5- bis 6-jährigen Kindergartenkindern im Kindergarten D durchzuführen. Gerade bei den Zeichenaktivitäten in den Lehr-Lern-Einheiten zeigte sich, dass die ganz jungen Kinder, im Alter von 3 bis 4 Jahren, oftmals beim Zeichnen daneben sitzen und warten, bis andere etwas gezeichnet haben, was sie dann nachzeichnen können. Oder die Kinder zeichnen Inhalte, die nicht mit der Aufgabenstellung in Zusammenhang stehen. Auch beim Software-Usability-Test (siehe Kapitel 9.2.6) und den anderen Usability-Evaluationen (siehe Kapitel 9.1.6 und 8.6.2.6) hat sich gezeigt, dass es ganz jungen Kindern, auf Grund ihrer noch fehlenden Vorerfahrung, oftmals noch etwas schwer fällt Inhalte, verbal wie auch non-verbal, zu kommunizieren. Die Kinder im Alter von 3 bis 4 Jahren profitieren folglich als Lernende am wenigsten von der Rolle als Gestalterinnen und Gestalter, da sie diese nicht vollends einnehmen können. Das exemplarische Endprodukt der Technikbildungsmaßnahme in der Informatik, ein Prototyp einer grafischen Benutzeroberfläche einer Mikrowelt, ist jedoch schon auch von 3- bis 4-jährigen Kindern zu evaluieren. Somit können hier die ganz jungen Kinder im Alter von 3 bis 4 Jahren Erfahrungen als Testerinnen und Tester sammeln.

Die Durchführung des Child-Centered-Design-Prozesses mit den 5- bis 6-jährigen Kindern erfolgt in vier Phasen, die insgesamt fünf Vormittage umfassen. Zu Beginn jeder einzelnen Session sowie auch während der Sessions erörtert die Autorin den teilnehmenden Kindern die Vorgehensweisen, die einzunehmenden Rollen und deren Einsatzzweck sowie die eingesetzten Methoden. Ebenso stellt die Verfasserin der gegenständlichen Arbeit zu Beginn stets das zu erreichende Ziel der jeweiligen Session vor.

Die erste Session beinhaltet das gemeinsame Zeichnen von Symbolen (Kinder in der Rolle als Gestalterinnen und Gestalter). Die Autorin ergänzt die Symbole im Anschluss um eigene Entwürfe und druckt jedes Symbol auf eine Karte. In den folgenden zwei Sessions wendet die Verfasserin der gegenständlichen Arbeit ein Card Sorting Verfahren an, bei dem die Kinder die Symbole den ihrer Meinung nach durch die Symbole repräsentierten Bedeutungen zuordnen (Kinder in der Rolle als Informantinnen und Informanten, siehe Kapitel 5.2.1). Abschließend sollen die Kinder für jede Bedeutungskategorie abstimmen, welches der vorgeschlagenen Symbole ihnen am besten gefällt. Mit diesen Erkenntnissen entwickelt die Autorin den Prototyp 1, den die Kinder anschließend in der Session 4 testen (Kinder in der Rolle als Testerinnen und Tester). Die in dieser Session auftretenden (schweren) Usability-Fehler behebt die Verfasserin der gegenständlichen Arbeit mit dem Prototyp 2 und unterzieht selbigen in Session 5 nochmals einem Test durch die Kinder.



Abbildung 31: Ablauf Child Centered Design

Mit Einverständnis der beteiligten Personen erfolgt zur Dokumentation der Geschehnisse die Aufzeichnung aller Aktivitäten per Video und die digitale Archivierung der einzelnen Ergebnisse der Sessions.

10.2 Symbole gestalten

Die erste Einheit (Session 1) diente zum Zeichnen und Entwickeln von Symbolen für einen Prototyp einer grafischen Benutzeroberfläche einer Mikrowelt zum Erlernen der Programmierung. An dieser Design-Session nahmen zehn Kinder im Alter von 5 bis 6 Jahren teil. Zu Beginn der Einheit fand eine gegenseitige Vorstellungsrunde statt, um sich miteinander vertraut zu machen. Des Weiteren stellte die Verfasserin der gegenständlichen Arbeit die von den Kindern einzunehmenden Rollen der Gestalterinnen und Gestalter sowie der Informantinnen und Informanten vor. Darüber hinaus skizziert sie den Ablauf und das Ziel der ersten Phase „Zeichnen“. Die Kinder lernen dadurch die Vorgehensweise zur Gestaltung von Symbolen für grafische Benutzeroberflächen kennen.

Auf diesen Einstieg folgte ein Frage-Antwort-Spiel rund um die bereits gewonnenen Erfahrungen der Kinder in der Computernutzung. Im Zuge dessen brachte die Autorin den Kindern auch den Begriff „Symbol“ näher, so dass alle wussten, wovon gesprochen wird und die aktive Arbeit an der Gestaltung von Symbolen beginnen konnte. Dazu teilte die Verfasserin der gegenständlichen Arbeit die Kinder in drei Gruppen zu je drei bzw. vier Kindern ein.

In den Gruppen fand das Zeichnen und Entwickeln von Symbolen für folgende Befehle, Eigenschaften oder Buttons statt:

- a) Aufgabenauswahl
- b) Programm beenden
- c) Hilfe
- d) Einen Schritt vorwärts gehen
- e) Einen Schritt rückwärts gehen
- f) Nach links drehen
- g) Nach rechts drehen
- h) Roboter Start
- i) Roboter Stopp
- j) Löschen
- k) Pause
- l) Aufgabe geschafft
- m) Aufgabe nicht geschafft

Die Kinder zeichneten die Symbole mit Farbstiften auf Kartonkärtchen, in der Größe 23,5 cm mal 10,5 cm. Die Betreuungspersonen versahen die Kärtchen auf der Rückseite mit den jeweiligen Begriffen, welche durch die Symbole dargestellt werden sollten, damit eine nachträgliche Zuordnung möglich war und nahmen darüber hinaus stellenweise Ergänzungen direkt beim Symbol mittels Schrift vor. Dieses Vorgehen, welches den Kindern kommuniziert wurde, vermittelt diesen die Grundlagen einer systematischen Dokumentation im Prozess.

10.2.1 Gruppe 1

In der ersten Gruppe zeichneten drei Mädchen und ein Junge jeder für sich einzeln an den Symbolen (siehe Anhang A6.1). Da die Kinder anfangs noch viel miteinander redeten und sehr unsicher waren was zu tun ist, halfen die Betreuungspersonen den Kindern und boten mit Fragestellungen sowie Hinweisen Unterstützung. Den Kindern war es so möglich, selbst zu erleben, wie mit Fragestellungen Unterstützung geleistet werden kann ohne das Ergebnis vorweg zu nehmen. Im Verlauf der Zeit zeigten sich die Kinder immer interessierter an dieser Zeicheneinheit und waren sehr aufmerksam sowie mit Konzentration bei der Sache.

Nicht alle Kinder konnten sich unter den Begriffen etwas vorstellen und so entwarfen nicht alle Kinder für alle Begriffe entsprechende Symbole. Es zeigte sich auch, dass einige Kinder

in Kooperation mit anderen Kindern ähnliche Symbole zeichneten, oder wenn sie gar nicht wussten was sie zu einem Begriff zeichnen sollten, ihrer Kreativität freien Lauf ließen.

10.2.2 Gruppe 2

Die zweite Gruppe bildeten drei Kinder, ein Mädchen und zwei Jungen, welche ebenfalls Symbole für die oben genannten Kategorien entwarfen (siehe Anhang A6.2). Auch hier zeigte sich, dass nicht für alle Begriffe entsprechende Symbole von den Kindern entwickelt werden konnten, Ähnlichkeiten vorhanden sind und die Aufgabenstellungen sehr kreativ gelöst wurden.

Interessant zu beobachten war in der zweiten Gruppe, dass sich die Kinder unter den Begriffen „Aufgabenauswahl“ oder „Programm beenden“ gar nichts vorstellen konnten und auch bei dem Begriff „Hilfe“ fiel es ihnen schwer, ein entsprechendes Symbol zu zeichnen. Ebenso interessant war die Erkenntnis, dass die für Erwachsene im Computergebrauch üblichen Farben wie Rot für Stopp oder Abbruch und Grün für Ok oder Start, bei den Kindern wenig bis gar keine Bedeutung hatten. Man beachte hierzu ein rotes Hakerl für Stopp (das Hakerl wird heutzutage eher für Bestätigen, OK o.ä. eingesetzt) oder das grüne Kreuzerl für Stopp (wobei das Kreuzerl eher für Abbrechen, Schließen oder Beenden eingesetzt wird).

10.2.3 Gruppe 3

In der dritten Gruppe waren wiederum drei Kinder aktiv, wobei alle drei Kinder bereits im Vorjahr an den Studien zum Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ und den Aktivitäten rund um den Bodenroboter Bee-Bot und die korrespondierende Softwaresimulation desselben Focus on Bee-Bot 1 bzw. 2 teilgenommen hatten und somit über entsprechendes Vorwissen verfügten (siehe Anhang A6.3). Die Verfasserin der gegenständlichen Arbeit stufte die Kinder dieser Gruppe daher als Expertinnen und Experten ein und bat sie ebenso die gewünschten Symbole zu zeichnen.

Auch bei der Gruppe der Expertinnen und Experten zeigte sich, dass es den Kindern schwerfiel, die Begriffe Aufgabenauswahl und Programm beenden in Symbole zu fassen – obwohl sie die Software Focus on Bee-Bot 1 bzw. 2 bereits kennengelernt hatten.

Aus der teilnehmenden Beobachtung der Autorin an der Session 1 – Zeichnen der Symbole zeigt sich informell, dass die teilnehmenden Kinder im Kindergartenalter sehr motiviert und aufmerksam mitgearbeitet haben. Die Kinder waren über den gesamten Zeitraum in ihrer Gruppe mit Elan und Eifer dabei und haben Symbole gestaltet. Zwischendurch mussten vereinzelt Kinder motiviert werden um weitere Symbole zu zeichnen, aber es war kein Kind dabei, das die Zeichenaktivitäten vorzeitig abgebrochen oder sich gegen das Zeichnen ausgesprochen hat.

10.2.4 Eigene Symbole

Die Verfasserin der gegenständlichen Arbeit skizzierte ebenso für fast jede Bedeutung zumindest ein Symbol, siehe Anhang A6.4. Die Symbole sind natürlich vor dem eigenen Erfahrungshintergrund und der Vermutung, welche Symbole Kinder eher ansprechen, entstanden.

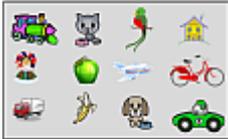
Die Autorin skizzierte nicht für alle Bedeutungen Symbole, da die Bedeutungen „rückwärts“ und „Aufgabe nicht geschafft“ invers zu „vorwärts“ und „Aufgabe geschafft“ zu sehen sind.

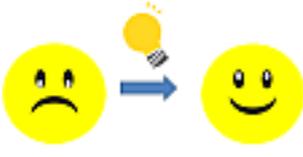
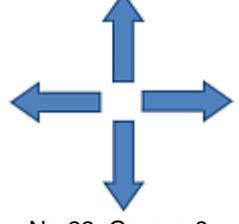
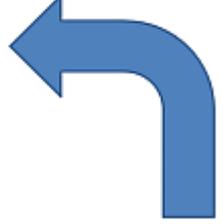
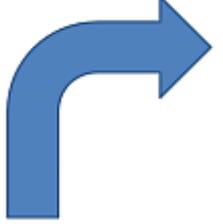
10.2.5 Symbolauswahl für Card Sorting

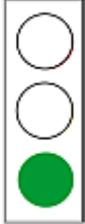
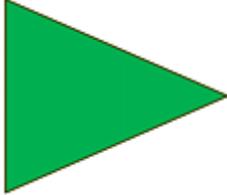
Nach der Session 1 in der Phase „Zeichnen“ gestaltete die Verfasserin der gegenständlichen Arbeit die Symbole, die durch ihre Ausgestaltung in ihrer Form klar identifizierbar waren, am Computer nach um erste mögliche Schaltflächen für den als Lernobjekt exemplarisch zu erstellenden Prototyp zu erzeugen, scannte einige andere Symbole ein und verarbeitete sie so weiter. Zur Überprüfung, ob Kinder mit Symbolen, welche nach Ansicht der Autorin eine geringe bis gar keine Aussagekraft hatten, im Kontext einer grafischen Benutzeroberfläche

einer Mikrowelt zum Erlernen von Programmierung arbeiten und diese effizient nutzen können bzw. ob Kinder diesen Symbolen überhaupt einer Bedeutung zuordnen können, zeichnete die Autorin diese Symbole ebenso nach.

Zuletzt wählte die Verfasserin der gegenständlichen Arbeit aus den 67 bestehenden Symbolen 54 Symbole aus, welche die am adaptierten Child-Centered-Design-Prozess teilnehmenden Kinder im Card Sorting Verfahren ihren Bedeutungen zuordnen sollten. Die Autorin nummerierte die Symbole durch, um sie identifizierbar zu machen und erfasste zusätzlich die Urheber- bzw. Ideengeber-Gruppe.

a) Aufgabenauswahl		
 <p>Nr. 28, Gruppe 3</p>	 <p>Nr. 47, Autorin</p>	 <p>Nr. 48, Autorin</p>
 <p>Nr. 49, Autorin</p>	 <p>Nr. 50, Autorin</p>	 <p>Nr. 51, Autorin</p>
b) Programm beenden		
 <p>Nr. 3, Gruppe 1</p>	 <p>Nr. 30, Autorin</p>	
c) Hilfe		
 <p>Nr. 1, Gruppe 1</p>	 <p>Nr. 5, Gruppe 1</p>	 <p>Nr. 20, Gruppe 2</p>
 <p>Nr. 39, Autorin</p>	 <p>Nr. 40, Autorin</p>	 <p>Nr. 41, Autorin</p>

 <p>Nr. 43, Autorin</p>	 <p>Nr. 44, Gruppe 3, Autorin</p>	
<p>d) Einen Schritt vorwärts gehen</p>		
 <p>Nr. 36, Autorin</p>	 <p>Nr. 6, Gruppe 1; Symbol für d und e</p>	 <p>Symbol 13, Gruppe 2; Symbol für d, e, f und g</p>
<p>e) Einen Schritt rückwärts gehen</p>		
 <p>Nr. 22, Gruppe 3; Symbol für d, e, f und g</p>		
<p>f) Nach links drehen</p>		
 <p>Nr. 31, Autorin</p>	 <p>Nr. 4, Gruppe 1; Symbol für f und g</p>	
<p>g) Nach rechts drehen</p>		
 <p>Nr. 35, Autorin</p>		
<p>h) Start</p>		
 <p>Nr. 16, Gruppe 2</p>	 <p>Nr. 18, Gruppe 2</p>	 <p>Nr. 21, Gruppe 2</p>

 <p>Nr. 29, Autorin</p>	 <p>Nr. 42, Autorin</p>	 <p>Nr. 46, Autorin</p>
<p>i) Stopp</p>		
 <p>Nr. 11, Gruppe 2</p>	 <p>Nr. 12, Gruppe 2</p>	 <p>Nr. 17, Gruppe 2</p>
 <p>Nr. 34, Autorin</p>		
<p>j) Löschen</p>		
 <p>Nr. 2, Gruppe 1</p>	 <p>Nr. 19, Gruppe 2</p>	 <p>Nr. 23, Gruppe 3</p>
 <p>Nr. 26, Gruppe 3</p>	 <p>Nr. 52, Gruppe 3, Autorin</p>	 <p>Nr. 53, Autorin</p>
 <p>Nr. 54, Autorin</p>		

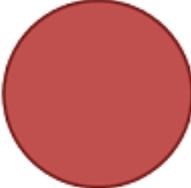
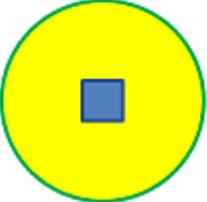
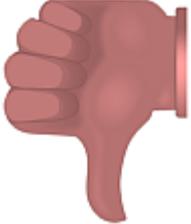
k) Pause		
 Nr. 7, Gruppe 2	 Nr. 14, Gruppe 2	 Nr. 15, Gruppe 2
 Nr. 24, Gruppe 3	 Nr. 45, Autorin	
l) Aufgabe geschafft		
 Nr. 8, Gruppe 2	 Nr. 9, Gruppe 2	 Nr. 27, Gruppe 3
 Nr. 37, Autorin	 Nr. 38, Autorin, Gruppe 1	
m) Aufgabe nicht geschafft		
 Nr. 10, Gruppe 2	 Nr. 25, Gruppe 3	 Nr. 32, Autorin
 Nr. 33, Autorin, Gruppe 1		

Tabelle 17: Ausgewählte Symbole für Card Sorting

10.3 Card Sorting - Symbole zuordnen

Die Sessions 2 und 3 standen ganz im Zeichen der Phase des „Card Sortings“. In den „Card Sorting“-Sessions sollten jene Kinder, welche zuvor die Symbole gezeichnet hatten, in Einzelsettings die ausgewählten Symbole den jeweiligen Bedeutungen zuordnen. Dazu wählte die Verfasserin der gegenständlichen Arbeit einen Closed Card Sorting Ansatz (siehe Kapitel 5.2.4), bei dem die Kinder die Symbole vorgegebenen Bedeutungen zuordnen sollen. Die Kinder lernen in der Rolle der Informantinnen und Informanten sowie der Testerinnen und Tester in der Phase „Card Sorting“ die Interpretationsmöglichkeiten von Symbolen, die Bewertung von Zuordnungen von Symbolen zu Begriffen sowie die Methodik und den Nutzen des Card Sortings kennen.

10.3.1 Setting und Durchführung

Im Vorfeld schrieb die Autorin die Bedeutungskategorien auf Täfelchen und ordnete diese am Boden nebeneinanderstehend an (siehe Abbildung 32). Die Symbole selbst druckte die Verfasserin der gegenständlichen Arbeit in Farbe auf A5 Kartonkarten und versah diese auf der Rückseite mit einer fortlaufenden Nummer, damit sie die Karten später wieder eindeutig identifizieren konnte. Diese Vorbereitungsarbeiten und die Grundlagen der Dokumentation der nachfolgenden Aktivitäten erklärte sie zu Beginn den teilnehmenden Kindern.

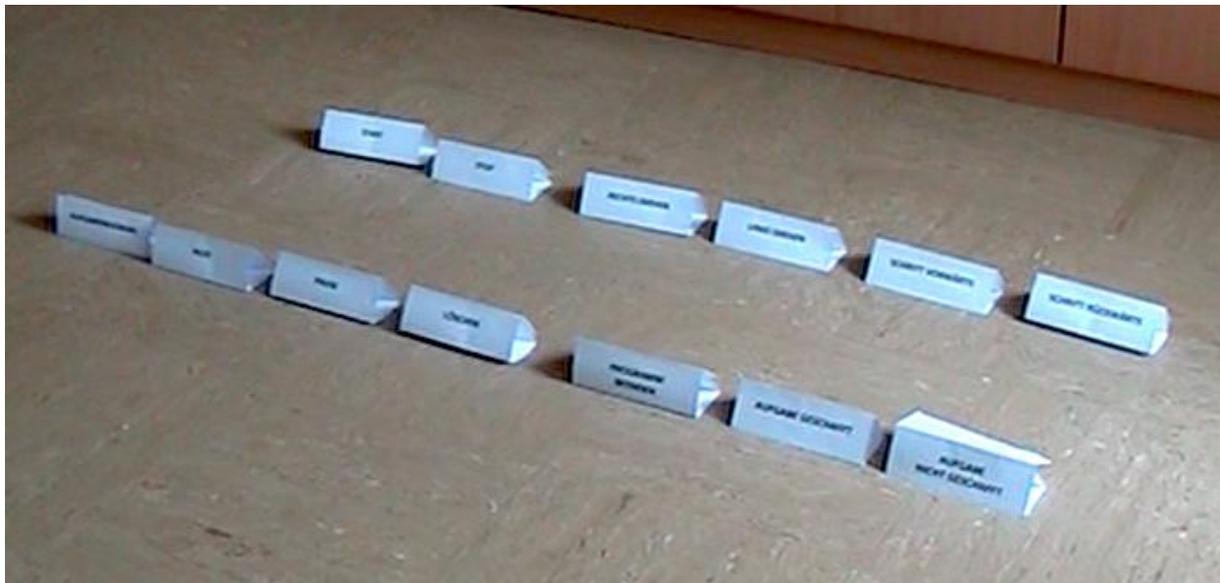


Abbildung 32: Anordnung Bedeutungskategorien

Ebenso besprach die Verfasserin der gegenständlichen Arbeit mit den Kindern den Ablauf, Einsatzzweck und die Ziele des Card Sortings zur Unterstützung der Gestaltung von grafischen Benutzeroberflächen sowie ihre Rolle als Informantinnen und Informanten sowie als Testerinnen und Tester in dieser Phase.

Anschließend forderte die Autorin die Kinder nacheinander in Einzelsettings auf die einzelnen Karten mit den Symbolen den Bedeutungen zuzuordnen. Da die Kinder im Alter von 5 bis 6 Jahren noch nicht lesen konnten, las die Autorin die Kategorien immer in der gleichen Reihenfolge vor und anschließend konnte das Kind sich eine entsprechende Kategorie für die Bedeutung des Symbols aussuchen. Im Verlauf der ersten Durchführung stellte sich heraus, dass dem ersten Kind die Zuordnung sehr schwerfiel, da es die einzelnen Begriffe / Bedeutungen verwechselte. Das Kind (und jedes weitere) bekam einen kleinen Ball in die Hand, den es werfen konnte, wenn der für das Kind passende Begriff genannt wurde. Wenn das Kind keine Kategorie wählen konnte, legte die Autorin die Karte mit dem betreffenden Symbol zur Seite (siehe Abbildung 33). Waren alle Karten zugeordnet, erfolgte ein weiterer Versuch, die Symbole der Karten des zur Seite gelegten Stapels zuzuordnen. Konnte das jeweilige Kind das Symbol einer Karte wiederum keiner Bedeutung zuordnen,

klassifizierte es die Verfasserin der gegenständlichen Arbeit für dieses Kind als „nicht zugeordnet“.

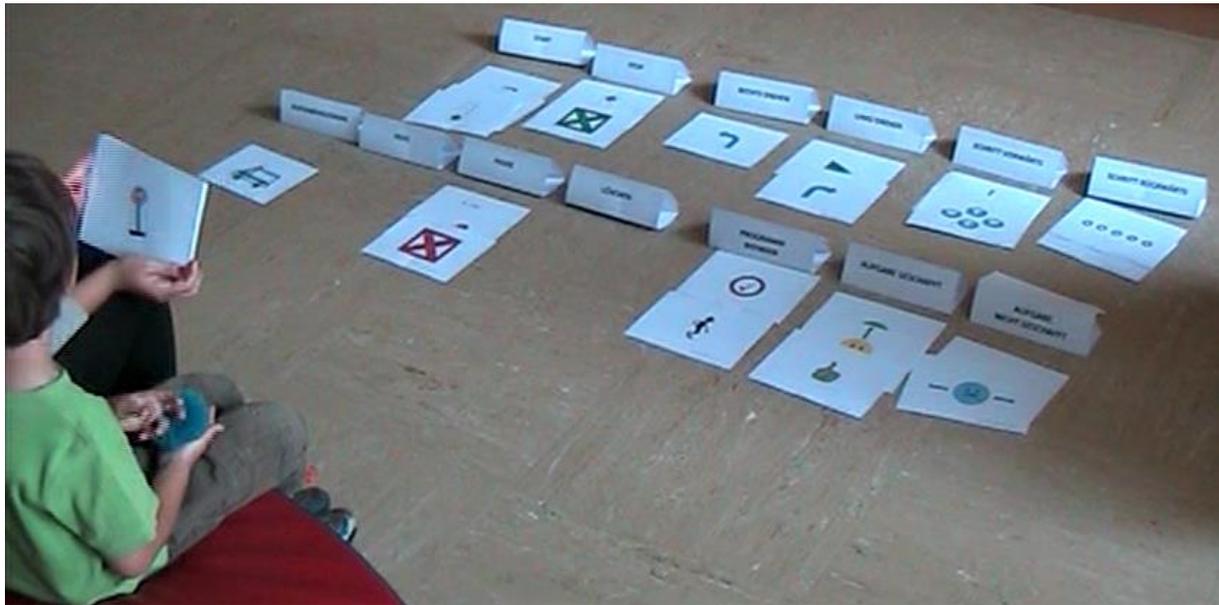


Abbildung 33: Zuordnung der Symbole zu den Bedeutungen

a) Aufgabenauswahl	
b) Beenden	
c) Hilfe	
d) Vorwärts	
e) Rückwärts	
f) Links	
g) rechts	
h) Start	
i) Stopp	
j) Löschen	
k) Pause	
l) geschafft!	
m) nicht geschafft	

Abbildung 34: Erfassungsbogen für die Zuordnung der Symbole zu Bedeutungen durch Kind

Nachdem ein Kind alle Karten mit den Symbolen zu den seiner Meinung nach passenden Bedeutungen zugeordnet hatte, erfasste die Autorin die Symbole mittels der Nummer auf der Rückseite der Karte auf einem Erfassungsbogen (siehe Abbildung 34) zu der jeweiligen Bedeutung. Dabei trug die Verfasserin der gegenständlichen Arbeit die Kartenummer in die grüne Zeile ein, wenn das Symbol zu Beginn schon für diese Bedeutung vorgesehen war –

also ein Kind oder die Autorin das Symbol explizit für diese Bedeutung zeichnete. In roten Zeilen erfasste die Autorin pro Bedeutung jeweils die Karten mit den Symbolen, welche nicht für diese Bedeutung gezeichnet wurden.

Nach der Erfassung der Zuordnung sammelte die Verfasserin der gegenständlichen Arbeit die Karten mit den Symbolen ein und mischte sie durch um zu vermeiden, dass im unwahrscheinlichen Fall das nächste Kind eine gleiche Zuordnung wie das vorherige Kind vornimmt. Die Notwendigkeit dieses Vorgehens, um als Forscherin und Forscher aussagekräftige Ergebnisse zu bekommen, kommunizierte die Autorin den Kindern.

Den Abschluss der Card Sorting Aktivitäten leitete die Autorin mit einem Gespräch über die Priorisierung von Symbolen und Möglichkeiten zur Ermittlung derselben ein.

Anschließend bat die Verfasserin der gegenständlichen Arbeit die Kinder, wieder in Einzelsettings, jene Symbole zu bewerten, welche ihnen am besten gefallen hatten. Die Symbole ordnete die Autorin zur Bewertung ihren zu Beginn angedachten Bedeutungen zu und forderte die Kinder auf mit farbigen Punkten, 13 Stück pro Kind (das entspricht einem Punkt pro Kategorie), das Symbol, welches ihnen am besten in einer Kategorie gefallen hat, zu kennzeichnen. Nachdem ein Kind all seine Punkte verteilt hatte, klebte die Verfasserin der gegenständlichen Arbeit die Punkte auf die Rückseite der jeweiligen Karten, um eine Beeinflussung der nachfolgenden Kinder in ihrer Meinung durch sichtbar aufgeklebte Punkte zu verhindern.

Zur Auswertung der Card Sorting Aktivitäten und der Bewertung der Symbole nach Gefallen, konsolidierte die Autorin die Daten und bereitete sie in einer Zusammenfassungstabelle auf. Dazu erfasste sie pro Symbol, zu welcher Bedeutung das Symbol ursprünglich zugeordnet wurde, wie oft die Kinder es richtig und wie oft falsch oder gar nicht zuordneten und wie viele Punkte für Beliebtheit (gefällt mir) die Kinder für das Symbol vergaben.

10.3.2 Endergebnis

Am Card Sorting Verfahren nahmen neun Kinder im Alter von 5 bis 6 Jahren teil, davon waren vier Kinder Mädchen und fünf Kinder Jungen. Die kürzeste Card Sorting Session dauerte 12:30 Minuten und die längste 21:45 Minuten, exkl. der Dokumentation der Zuordnungen. Die Länge der Sessions hing vor allem mit der Entscheidungsfreudigkeit der Kinder zusammen und mit allfälligen Ablenkungen von außen (Kindergartenpädagoginnen kamen zwischendurch vorbei, draußen spielten andere Kinder und es war lauter, ...). So nahm die Durchführung des Card Sortings zwei Vormittage in Anspruch.

Tabelle 18 zeigt die konsolidierten Ergebnisse des Card Sortings und die Bewertung der Beliebtheit der Symbole. Die jeweils fünf größten Ergebnisse pro Spalte kennzeichnete die Autorin grün und die fünf kleinsten Ergebnisse rot. Darüber hinaus sortierte die Verfasserin der gegenständlichen Arbeit die Tabelle pro Kategorie nach der größten Anzahl richtig zugeordneter Symbole, nach der größten Beliebtheit, nach der geringsten Anzahl an falschen Zuordnungen und der geringsten Anzahl keiner Zuordnung. Durch diese Sortierung steht jeweils pro Kategorie oben als erstes jenes Symbol, welches am meisten richtig zugeordnet wurde, am beliebtesten ist, am wenigsten falsch und am wenigsten gar nicht zugeordnet wurde.

Bei Symbolen, bei welchen die Auswertung nicht direkt zu einem sinnvollen Ergebnis führte, sammelte die Verfasserin der gegenständlichen Arbeit noch weitere Informationen. So z.B. bei Symbol Nr. 30 (gezeichnet für Programm beenden), das zwar nur eine richtige Zuordnung hatte, aber bei den Kindern mit sieben Nennungen zu den beliebtesten von allen Symbolen zählte. Die Autorin analysierte zu welcher Bedeutung die Kinder das Symbol am häufigsten falsch zugeordnet hatten. Symbol Nr. 30 ordneten die Kinder eher zu Stopp, Löschen und nicht geschafft zu. Der Einsatz des Symbols muss daher sehr wohl überlegt werden, um Missverständnissen vorzubeugen.

Aus der teilnehmenden Beobachtung der Autorin lässt sich informell ableiten, dass die teilnehmenden Kinder sehr interessiert an den Details des Designprozesses waren und

gerne und gewissenhaft mitarbeiteten. Sie waren konzentriert, aufmerksam und stellten Fragen zur Vertiefung ihres Verständnisses. Bei keinem teilnehmenden Kind musste der Card Sorting Prozess abgebrochen werden, kein Kind musste zur Mitarbeit überredet bzw. ermahnt werden. Kritisch könnte man die Menge der zuzuordnenden Symbole betrachten, wo die Geduld der Kinder auf die Probe gestellt wurde. Wobei dies bei der praktischen Durchführung des Card Sortings zu keinen Problemen führte und sich die teilnehmenden Kinder als sehr kooperativ erwiesen.

Kinder können als Gestaltungspartnerinnen und Gestaltungspartner bei der Entwicklung von Symbolen mitwirken, sie sind mit Begeisterung dabei und freuen sich auf die Herausforderung. Ob die Kinder Erfahrungen in der Nutzung von Computern haben oder bereits mit ähnlichen Produkten (Software wie Hardware) gespielt haben, führt zu geringen bis gar keinen Unterschieden. Natürlich fiel es in diesem Evaluationssetting den Kindern der Gruppe 3 leichter, Symbole zu zeichnen, da sie auf eventuell noch vorhandenes Vorwissen zurückgreifen konnten und die gesuchten Bedeutungen eher kannten. Das zeigen die einzelnen Symbole welche die Kinder dieser Gruppe zeichneten. Kritisch anzumerken ist, dass die Kinder der Gruppe 3 möglicherweise voreingenommen agierten und bereits Gesehenes replizierten. Gänzlich neue Ideen, wie sie aus der Gruppe 1 und 2 kamen, erscheinen im ersten Moment kreativ und interessant, spielen aber in weiterer Folge nur eine kleine Rolle. Das zeigt das Ergebnis des Card Sortings. Kaum eine Symbol-Idee der Kinder selbst schaffte es in die finale Liste der einzusetzenden Symbole (siehe Tabelle 18 und Tabelle 19). Klar festzuhalten ist, dass die Ideen, welche die Kinder während des Zeichnens und Gestaltens einbringen und ihre Nachfragen eine sehr wichtige Inspirationsquelle sind (Kinder agieren hier als Informantinnen und Informanten), welche die Autorin bei der Gestaltung einer zielgruppengerechten Benutzeroberfläche nicht missen möchte.

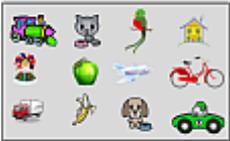
Kategorie	Bedeutung	Symbol nr.	richtig zugeord.	falsch zugeord.	nicht zugeord.	Beliebtheit
a	Aufgabenauswahl	49	7	0	2	3
a	Aufgabenauswahl	47	3	1	5	2
a	Aufgabenauswahl	48	3	3	3	2
a	Aufgabenauswahl	50	1	6	2	2
a	Aufgabenauswahl	51	1	2	6	1
a	Aufgabenauswahl	28	1	4	4	0
b	Beenden	30	1	7	1	7
b	Beenden	3	1	4	4	2
c	Hilfe	44	8	1	0	3
c	Hilfe	1	3	3	3	0
c	Hilfe	43	2	3	4	0
c	Hilfe	41	1	4	4	4
c	Hilfe	40	1	4	4	1
c	Hilfe	39	0	5	4	1
c	Hilfe	5	0	3	6	0
c	Hilfe	20	0	4	5	0
d	Vorwärts	13	8	0	1	8
d	Vorwärts	22	8	0	1	1
d	Vorwärts	36	6	2	1	4
d	Vorwärts	6	3	2	4	3
f	Links	31	7	2	0	2
f	Links	4	3	3	3	7
g	Rechts	35	7	2	0	7

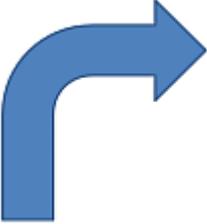
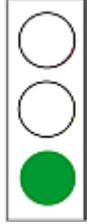
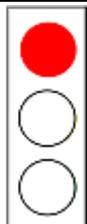
Kategorie	Bedeutung	Symbol nr.	richtig zugeord.	falsch zugeord.	nicht zugeord.	Beliebtheit
h	Start	29	9	0	0	3
h	Start	16	6	2	1	4
h	Start	42	1	4	4	2
h	Start	21	1	3	5	1
h	Start	18	1	5	3	0
h	Start	46	0	8	1	0
i	Stopp	11	8	0	1	4
i	Stopp	34	8	1	0	3
i	Stopp	17	2	4	3	1
i	Stopp	12	0	8	1	1
j	Löschen	53	8	1	0	4
j	Löschen	19	6	2	1	4
j	Löschen	54	5	1	3	0
j	Löschen	52	4	2	3	0
j	Löschen	2	1	3	5	1
j	Löschen	26	0	4	5	0
j	Löschen	23	0	6	3	0
k	Pause	45	8	1	0	1
k	Pause	14	7	2	0	0
k	Pause	24	1	6	2	7
k	Pause	7	1	7	1	1
k	Pause	15	0	4	5	0
l	geschafft!	38	8	1	0	3
l	geschafft!	37	7	2	0	2
l	geschafft!	27	6	2	1	3
l	geschafft!	9	4	4	1	1
l	geschafft!	8	4	5	0	0
m	nicht geschafft!	33	8	1	0	3
m	nicht geschafft!	32	7	2	0	4
m	nicht geschafft!	25	6	3	0	1
m	nicht geschafft!	10	4	4	1	1

Tabelle 18: Konsolidierte Ergebnisse Card Sorting und Bewertung gesamt

Ein detaillierter Blick auf die Ergebnistabellen, zum einen Tabelle 32 in Anhang A7, die die Ergebnisse nach fünf Kindern und zum anderen Tabelle 18 auf Seite 223, die die Ergebnisse aller Kinder beinhaltet, zeigt, dass sich die Ergebnisse von fünf auf neun Kinder kaum mehr geändert haben, lediglich im Detail verfeinert und im Gesamten gefestigt wurden. Dies belegt, dass, ähnlich zu Nielsen (1993), auch bereits fünf bis sechs Kinder im Alter von 5 bis 6 Jahren ausreichend sind, um Symbole und deren Bedeutungen zu evaluieren.

Die für den, im Zuge des adaptierten Child-Centered-Design-Prozesses zu erstellenden, Prototyp ausgewählten Symbole sind in Tabelle 19 dargestellt. Diese Auswahl traf die Autorin basierend auf den Endergebnissen der Card Sorting Aktivitäten und Bewertungen.

a) Aufgabenauswahl		
		
Nr. 49, Autorin		
b) Programm beenden		
		
Nr. 30, Autorin		
c) Hilfe		
		
Nr. 44, Autorin		
d) Einen Schritt vorwärts gehen		
		
Nr. 13, Gruppe 2; Symbol für d, e, f und g		
e) Einen Schritt rückwärts gehen		
f) Nach links drehen		
		
Nr. 4, Gruppe 1; Symbol für f und g		

g) Nach rechts drehen		
 Nr. 35, Autorin		
h) Start		
 Nr. 16, Gruppe 2	 Nr. 29, Autorin	
i) Stopp		
 Nr. 11, Gruppe 2	 Nr. 34, Autorin	
j) Löschen		
 Nr. 53, Autorin		
k) Pause		
 Nr. 24, Gruppe 3	 Nr. 45, Autorin	
l) Aufgabe geschafft		
 Nr. 38, Autorin		

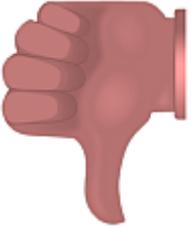
m) Aufgabe nicht geschafft		
 Nr. 32, Autorin	 Nr. 33, Autorin	

Tabelle 19: Finale Symbole nach Card Sorting und Bewertung

10.4 Prototyp

Die Verfasserin der gegenständlichen Arbeit entwickelte im Rahmen der Durchführung des adaptierten Child-Centered-Design-Prozesses mit Kindergartenkindern zur frühen Technikbildung in Informatik exemplarisch einen Prototyp einer grafischen Benutzeroberfläche einer Mikrowelt zum Erlernen der Programmierung für 3- bis 6-jährige Kinder unter Berücksichtigung der aus der Literatur gewonnenen Erkenntnisse (siehe Kapitel 4.1, 4.2 und 5.1).

Durch die Teilnahme am Prototyping, der vierten Phase im adaptierten Child-Centered-Design-Ansatz, können die Kinder die Rollen Testerinnen und Tester sowie Benutzerinnen und Benutzer erleben und kennenlernen. Im Zuge der Evaluierungs-Tätigkeiten lernen und üben sie die Verbalisierung ihrer Gedanken und Ideen zur Verbesserung von Produkten. Die anschließende Iteration der Entwicklung des Prototyps zeigt den teilnehmenden Kindern auf, wie ihr zuvor gegebenes Feedback in die Weiterentwicklung eines Produktes einfließt. In weiterer Folge lernen die Kinder einen exemplarischen Prototyping-Prozess kennen und dass Prototypen über unterschiedliche Ausgestaltungsgrade und Abstraktionen verfügen und so gleiche Funktionalitäten / Eigenschaften unterschiedlich repräsentiert werden können. Das Kennenlernen, die Vorstellung und das Erleben von Evaluierungstests mit ihren Fragebögen, Untersuchungsgegenständen und Dokumentationswerkzeugen, wie Audio- und Videoaufzeichnung oder Screenrecording, hilft den Kindern das Verständnis über die Evaluierung von Produkten auszuprägen. Zusätzliche Funktionen, die durch den Prototyp zur Verfügung gestellt werden, ermöglichen den Kindern erste Erfahrungen mit einem tutoriellen System als Hilfesystem zu sammeln und weitere Konzepte der Informatik, wie zum Beispiel den „Kürzesten Weg“, kennenzulernen.

Einleitend beschäftigt sich die Autorin in Kapitel 10.4.1 mit der Darlegung ihrer Überlegungen zur Gestaltung des Prototyps sowie der Dokumentation der Ausgestaltung des Paper Prototyps zum Klick-Prototyp in der ersten Version. Die Evaluierung der ersten Version des klickbaren Prototyps durch die Kinder ist in Kapitel 10.4.2 zu finden. Die aus der Evaluierung gewonnenen Erkenntnisse fließen in die Entwicklung der zweiten Version des Prototyps ein (siehe Kapitel 10.4.3), welche wiederum evaluiert wird (siehe Kapitel 10.4.4).

10.4.1 Entwicklung des Prototyps – Version 1

Die Entwicklung des exemplarischen Prototyps einer grafischen Benutzeroberfläche einer Mikrowelt zum Erlernen der Programmierung für 3- bis 6-jährige Kinder basiert auf den Vorerfahrungen der Autorin, der zuvor durchgeführten Literatuarbeit (siehe Kapitel 4.1, 4.2 und 5.1) und den gewonnenen Erkenntnissen aus der Phase „Zeichnen“ (siehe Kapitel 10.2) sowie der Phase „Card Sorting“ (siehe Kapitel 10.3).

10.4.1.1 Überlegungen und Grundlagen

Die zu erstellende Mikrowelt soll den Kindern die Möglichkeit bieten, erste Grundkonzepte der Programmierung zu erlernen und zu trainieren. Der Prototyp soll die Bedürfnisse und Anforderungen der Kinder decken. Angelehnt an bekannte Bodenroboter, wie zum Beispiel den Bee-Bot oder Constructa-Bot (siehe Kapitel 9.1.1), an Software zum Programmieren

lernen für Kinder, wie zum Beispiel die Software Focus on Bee-Bot 1 bzw. 2 (siehe Kapitel 9.2.1) und die in der Durchführung der Lehr-Lern-Einheiten zur Entwicklung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ gewonnenen Erfahrungen zum Erlernen der ersten Grundelemente der Programmierung, soll der Prototyp mit einem spielerischen Ansatz sowie einer einfachen Bedienung das Interesse an Programmierung weiter stärken bzw. wecken.

Die Autorin gestaltet mit dem Prototyp eine grafische Benutzeroberfläche einer Mikrowelt zum Erlernen der Programmierung, die es Kindern ermöglichen soll, übergeordnete Kompetenzen wie das Vorausplanen, das Erfassen von komplexen Zusammenhängen, sowie das Erstellen einfacher Algorithmen kennenzulernen und zu trainieren. Darüber hinaus wird das Erkennen und Wählen des kürzesten Weges (shortest path) gefördert – ein zentrales Konzept der Informatik. Dazu soll eine Spielfigur, in diesem Fall ein Schmetterling, auf einem Spielfeld zu bestimmten Symbolen fliegen. Die Steuerung des Schmetterlings erfolgt auf einfachere Weise als zum Beispiel in der Software Focus on Bee-Bot 1 bzw. 2 und ist stärker auf die Bedürfnisse der Kinder ausgerichtet. Die Autorin betrachtete daher die Implementierung einer Touch-Screen Oberfläche als obligatorisch und reduzierte auch die Anzahl der Bedienelemente auf ein absolutes Minimum. Da die Zielgruppe, 3- bis 6-jährige Kinder, in der Regel kaum lesen kann, verzichtete die Verfasserin der gegenständlichen Arbeit weitestgehend auf Text.

Die Implementierung eines einfachen Hilfesystems ist ebenso fixer Bestandteil, denn die Kinder sollen zunehmend komplexere Themen mit der Lernumgebung lösen können, ohne dass sie unbedingt von einer erwachsenen Betreuungsperson Hilfestellungen benötigen. Die Programmierung der Spielfigur erfolgt nach dem FIFO-Prinzip (First-In-First-Out), also der Befehl, welcher zuerst eingegeben wird, wird auch zuerst abgearbeitet.

10.4.1.2 Paper Prototyp

Die Entwicklung des Prototyps startete mit handgezeichneten Skizzen, welche die Umrisse und Grundstruktur der Benutzeroberfläche enthalten. Die Autorin erstellte ein erstes Scribble bzw. einen ersten Paper-Prototyp.

Dem ersten Paper-Prototyp sind die grundlegenden Bereiche und Elemente der Bedienoberfläche zu entnehmen. Am Paper-Prototyp 1 (siehe Abbildung 35) ist oberhalb des Rasters mit Symbolen die Aufgabenstellung zu erkennen. Die Spielfigur soll von ihrer aktuellen Position zum angezeigten Symbol fahren. Dieser Aufgabenstellungsbereich findet sich auch in den nächsten Paper-Prototypen (siehe Abbildung 36 und Abbildung 37) als zentrales Element, neben dem Raster mit den Symbolen, wieder.

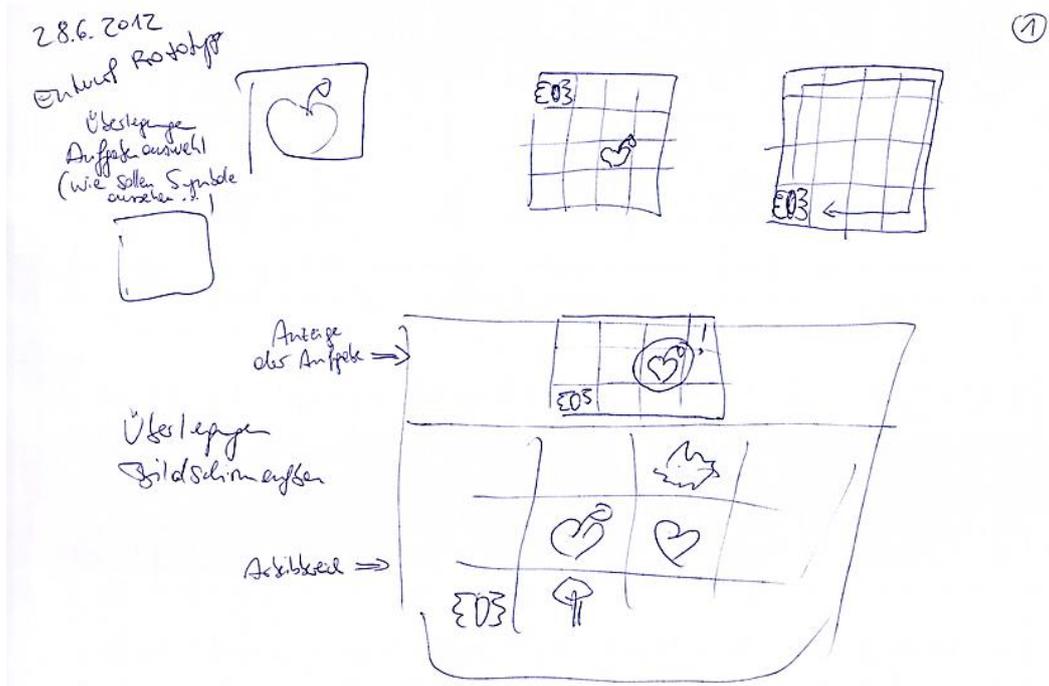


Abbildung 35: Paper-Prototyp 1

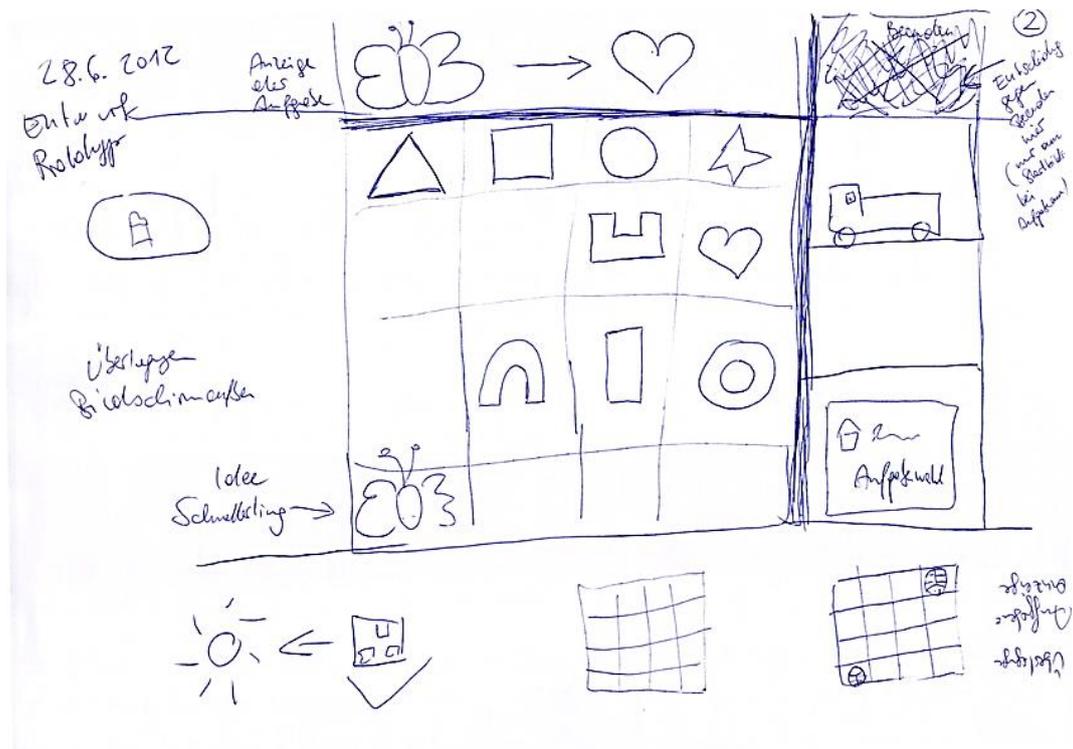


Abbildung 36: Paper-Prototyp 2

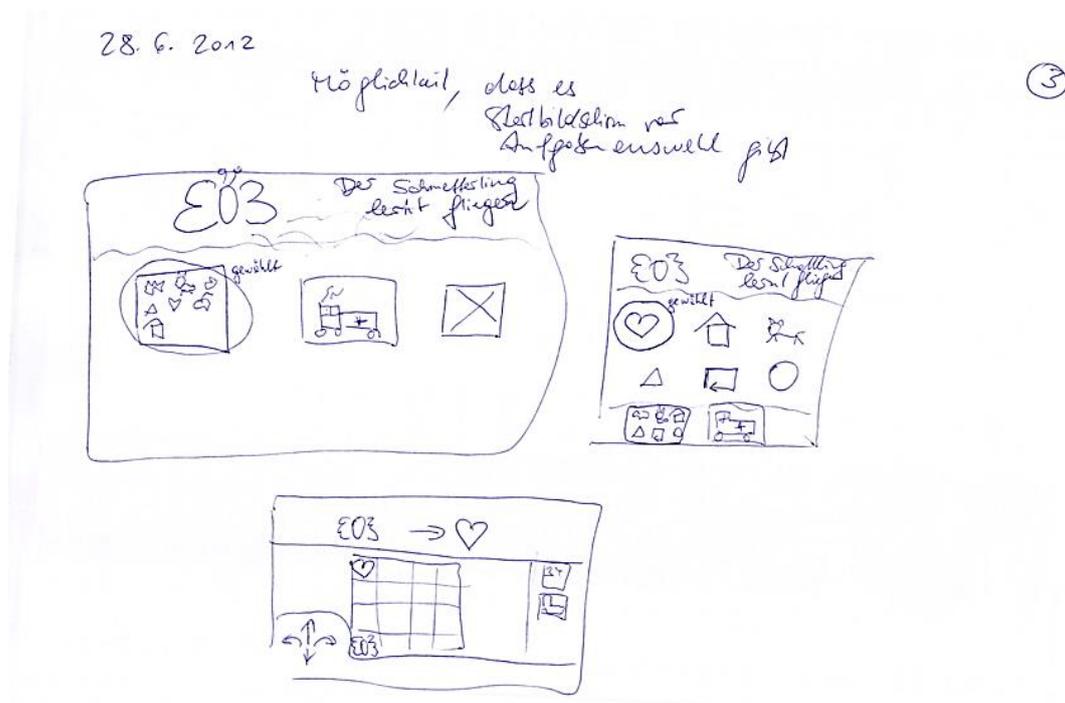


Abbildung 37: Paper-Prototyp 3

Der Raster mit den Symbolen wird als Arbeitsfläche bezeichnet. Jedes Feld beinhaltet ein unterschiedliches Symbol (siehe Abbildung 36). Da die Kinder der Zielgruppe in der Regel noch nicht lesen können und die Nutzung der Software auch weitere Kompetenzen trainieren könnte, wählte die Autorin hierzu im Prototyp geometrische Symbole, welche es zu erkennen und bei der Aufgabenlösung zu erreichen gilt. Im rechten Bereich sind erste Buttons in entsprechender Größe für eine Touch-Screen Bedienung zu erkennen, die eine Hilfe-Funktion (oben) und eine Aufgabenauswahl (unten) zur Verfügung stellen.

Im Paper-Prototyp 3 (siehe Abbildung 37) sind zwei Entwürfe für den Bereich der Aufgabenauswahl zu erkennen. In diesem speziellen Bereich bzw. Bildschirm sollen Kinder aus unterschiedlichen Aufgabenstellungen auswählen können. Damit die Kinder Hilfe in Anspruch nehmen können, wenn sie selbst nicht wissen, wie das System zu bedienen ist, ist auch ein Hilfe-Button (hier ein Symbol für einen Krankenwagen, siehe dazu auch die Ergebnisse des Card Sortings, Kapitel 10.3.2) vorhanden. In der unteren Skizze des Paper-Prototyps 3 ist eine erste vollständige Übersicht über die Benutzeroberfläche zu sehen. Im oberen Bereich befindet sich, wie schon erwähnt, die Aufgabenstellung, auf der rechten Seite die Buttons für Hilfe und Auswahl sowie in der Mitte die Arbeitsfläche mit dem Raster und den Symbolen. Links unten befinden sich die Schaltflächen zur Steuerung und Programmierung der Spielfigur sowie unterhalb der Arbeitsfläche der Befehlsbereich, in dem die eingegebenen Befehle nebeneinander dargestellt werden.

Bei der Evaluierung der Usability der Focus on Bee-Bot 1 bzw. 2 Software stellte sich unter anderem heraus, dass Kinder irritiert waren, wenn z.B. der Bee-Bot in entgegengesetzter Blickrichtung zu den Richtungs-Buttons im Navigationsbereich (und damit zur Blickrichtung der Kinder selbst) stand. Dies deckt sich mit Erkenntnissen aus der Literatur, siehe Kapitel 5.1.1.2. Wenn dazu die eingegebenen Befehle in der Befehlsliste ebenso ausgerichtet wie die Richtungs-Buttons im Navigationsbereich, nur weiter unten, angezeigt wurden, verwechselten die Kinder oftmals den Rückwärts-Pfeil mit einem „Runter-Fahren“-Befehl, der aus Blickrichtung des Kindes korrekt wäre, jedoch nicht aus Sicht der Programmierung des Bee-Bots. Daher achtete die Autorin bei der Gestaltung des Prototyps darauf, dass die Befehle nebeneinander angeordnet werden, um für die Kinder durch die natürliche

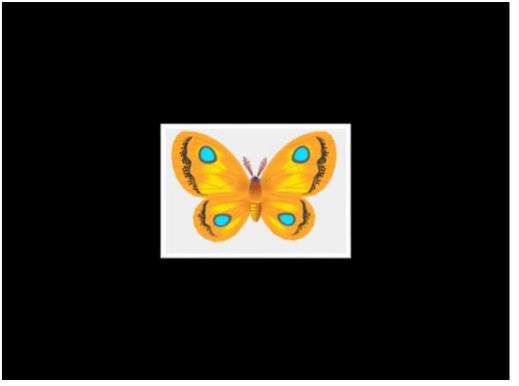
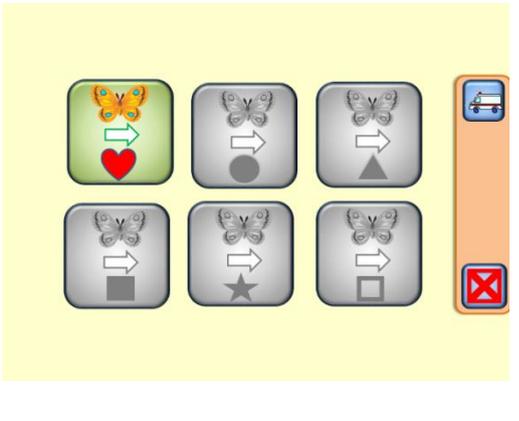
Leserichtung etwas mehr Klarheit bei der Interpretation der unterschiedlichen Perspektiven zu schaffen.

10.4.1.3 Ausgestaltung Prototyp – Version 1

Nach den skizzierten Entwürfen der Benutzeroberfläche gestaltete die Verfasserin der gegenständlichen Arbeit den ersten Prototyp der Benutzeroberfläche mit Microsoft® PowerPoint®. Der erste Prototyp ist klickbar, also tatsächlich als Demonstrations-Objekt nutzbar. Er speichert keine Daten und aktualisiert keine Zeitanzeigen, da der Prototyp aus einer Aneinanderreihung von Einzelbildern besteht, welche durch die Integration von interaktiven Flächen die Möglichkeit bieten, durch Klick auf eine spezielle Fläche ein zuvor spezifiziertes Bild anzeigen zu lassen. Der Prototyp ist daher in seiner Ausgestaltung zwischen einem Low-Fidelity und einem High-Fidelity Prototyp anzusiedeln.

Durch das Folienkonzept, welches Präsentationssoftware zu Grunde liegt, die umfangreichen Gestaltungsfunktionalitäten und die einfachen Bedienmöglichkeiten von Präsentationssoftware, die rasch zu einem Ergebnis führen, eignet sich diese Art von Software generell sehr gut zum Erstellen von Prototypen, welche über reine Skizzen hinaus gehen und schon möglichst reale Benutzeroberflächen darstellen sollen. Darüber hinaus liefert der Präsentationsmodus die Möglichkeit der Simulation der Oberfläche, da nur mehr die entsprechenden Bilder angezeigt und die Bedienelemente der Präsentationssoftware selbst ausgeblendet werden.

Die folgende Tabelle 20 zeigt die wichtigsten Bilder der Benutzeroberfläche mit einer entsprechenden Kurzbeschreibung. Die Buttons gestaltete die Autorin basierend auf den Ergebnissen des Card Sortings (siehe Kapitel 10.3.2) und integrierte sie entsprechend ihrer Bedeutung.

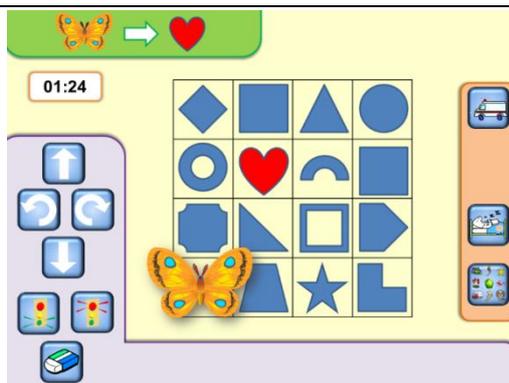
Bild des Prototyps	Kurzbeschreibung
	<p>Start:</p> <p>Für den Start der Evaluierung und des Durchklickens des Szenarios des Prototyps wird die Spielfigur als Start-Button auf schwarzem Hintergrund angezeigt.</p>
	<p>Aufgabenauswahl:</p> <p>Auf großen Buttons wird den Kindern die Möglichkeit zur Aufgabenauswahl gegeben, wobei mögliche Aufgabenstellungen grün hinterlegt sind (aktiv) und zurzeit nicht verfügbare Aufgabenstellungen grau hinterlegt sind (nicht aktiv).</p> <p>Die Aufgabenstellung selbst ist auf dem Button bereits mit abgebildet – hier soll die Spielfigur, der Schmetterling, zum roten Herz fliegen.</p> <p>Auf der rechten Seite, im orangen Bereich, ist oben die Taste für die Hilfe und unten die Taste für das Beenden zu finden.</p>



Hilfe:

Diese Bildschirmoberfläche bietet als Hilfestellung eine kleine Übersicht des eigentlichen Programms. Ein roter Kreis und der damit umrandete Pfeil markiert den zuerst einzugebenden Befehl um die oben im grünen Bereich angezeigte Aufgabenstellung zu lösen. Der grüne Pfeil vor der Spielfigur zeigt an, dass sich der Schmetterling in diese Richtung bewegen soll.

Der schwarze Hintergrund ist absichtlich leicht transparent, damit die Kinder erkennen können, dass sie, nachdem sie das Bild gesehen haben, durch Tippen gleich wieder auf die eigentliche Bildschirmoberfläche zurückkehren können.



Haupt-Bildschirmoberfläche:

Links oben, im grünen Bereich, ist die Aufgabenstellung stets verfügbar. Der Schmetterling soll zum roten Herz fliegen.

Darunter befindet sich eine Zeitanzeige, welche die Zeit bis zur Lösung misst. Die Zeitmessung ist im Prototyp nicht aktiv und nur durch einen Platzhalter repräsentiert.

In der Mitte, auf gelbem Hintergrund, befindet sich der Raster mit den Symbolen – der Arbeitsbereich. Hier können unterschiedliche Symbole pro Kästchen eingesetzt werden, um unterschiedliche Aufgabenstellungen zu ermöglichen. Im Prototyp nutzte die Autorin die blauen geometrischen Figuren und hob das Ziel jeweils rot hervor.

Der orange Bereich rechts ist kontextabhängig und beinhaltet stets die im Moment notwendigen Elemente. Der unterste Button in diesem Bereich bietet immer die Funktionalität „zurück“ an. In diesem Fall geht es mit dem untersten Button zurück zur Aufgabenauswahl. Um die Kinder nicht durch weitere Pfeile zu irritieren, verwendete die Verfasserin der gegenständlichen Arbeit hier das Symbol für Aufgabenauswahl.

Darüber befindet sich der Pause-Button, falls das Kind während der Lösung einer Aufgabe eine Pause machen möchte. In diesem Fall würde die Zeit gestoppt werden.

Links, im lila Bereich, befindet sich zuerst der Navigationsbereich als Steuermöglichkeit für den Schmetterling. Neben einem Schritt vorwärts und rückwärts finden sich hier auch die Buttons für Links-Drehen und Rechts-Drehen (um 90 Grad).

Darunter befinden sich zwei Ampel-Buttons: Links mit grün für Start, damit der Schmetterling die einprogrammierte Befehlsfolge abzuarbeiten beginnt. Rechts mit rot für Stopp, damit der Schmetterling die Abarbeitung der einprogrammierten Befehlsabfolge stoppt. Darunter befindet sich ein Radiergummi für Löschen – damit können einzelne Befehle wieder aus

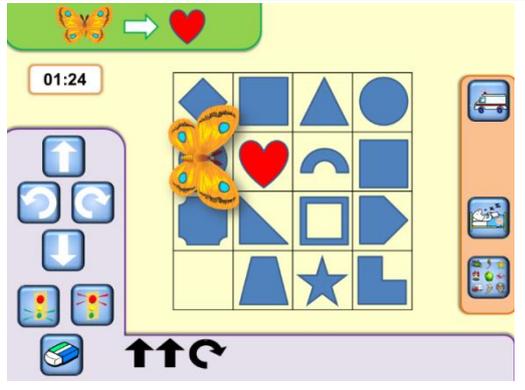
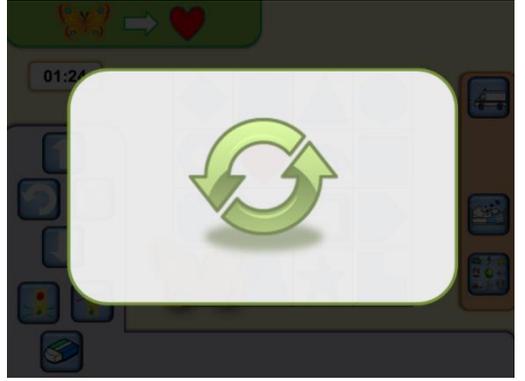
	<p>der Befehlsliste gelöscht werden.</p> <p>Der Bereich in lila unterhalb des Arbeitsbereichs dient der Auflistung der eingegebenen Abfolge von Richtungs-Befehlen.</p>
	<p>Bildschirmoberfläche während der Bearbeitung:</p> <p>In diesem Fall ist die Befehlsliste, in Form einer Leiste, bereits mit drei Befehlen (Vorwärts, Vorwärts, Rechts-Drehen) gefüllt und Start wurde gedrückt. Daraufhin ist der Schmetterling einen Schritt Vorwärts, einen Schritt Vorwärts und eine Rechts-Drehung um 90 Grad gefahren.</p>
	<p>Aufgabe gelöst:</p> <p>Wenn die Aufgabe gelöst wurde, werden über der Arbeitsfläche der Daumen nach oben und das Wort „Super“ angezeigt.</p> <p>Auch hier ist diese Rückmeldung, wie die Hilfe, als Overlay gestaltet, so dass ein Klick, an beliebiger Stelle auf dem Bildschirm, wieder dafür sorgt, dass es mit der Aufgabenauswahl weitergeht.</p> <p>In der Befehlsliste sind nun vier Befehle enthalten, die den Schmettling zum Ziel, dem roten Herz, geführt haben.</p>
	<p>Pause:</p> <p>Der Pause-Bildschirm ist ähnlich zur Hilfe gestaltet. Während der Pause läuft die Uhr nicht weiter. Der transparente Hintergrund soll wieder dazu einladen, durch einen einfachen Klick an beliebiger Stelle auf den Bildschirm, die Pause zu beenden.</p>
	<p>Aufgabe nicht geschafft:</p> <p>Wurde die gestellte Aufgabe falsch gelöst oder ohne Eingabe von Befehlen die grüne Ampel für „Start“ gedrückt, erscheint der Wiederholungspfeil. Dieser Pfeil wurde gewählt, da ein „Leider nein“ oder ein roter Daumen nach unten für die Motivation weniger dienlich sein könnten, als ein grüner, motivierender Pfeil, der zu einem weiteren Versuch auffordern soll.</p>

Tabelle 20: Übersicht Prototyp 1

10.4.2 Evaluation Prototyp Version 1

Zur Evaluation des Prototyps stellte die Verfasserin der gegenständlichen Arbeit die gewünschte Demo-Funktionalität, der Schmetterling soll von seinem Ausgangspunkt zum roten Herz fliegen, in Form einer Abfolge von einzelnen Präsentationsfolien her. Die finale Präsentation und somit der Prototyp umfasst 51 Präsentationsfolien und deckt die Varianten zur Lösung der Aufgabe ab, die die Kinder mit den kürzest möglichen Wegen finden können.

Zu Beginn des Testszenarios erörterte die Verfasserin der gegenständlichen Arbeit dem jeweils anwesenden Kind die Ziele der Evaluierung, die vom Kind einzunehmende Rolle und die generelle Vorgehensweise bei der Evaluierung.

Weiter vermittelte die Autorin den teilnehmenden Kindern die Unterschiede zwischen einem klickbaren Prototyp, der simuliert wird, und einem tatsächlich funktionalem Software Produkt. Die Autorin führte als Ergänzung aus, dass Testerinnen und Tester einen erheblichen Beitrag zur Erkenntnissammlung leisten und der Erforschung von Neuem dienlich sind. Die Kinder können durch die Teilnahme an der Evaluierung die Tätigkeiten einer Forscherin und eines Forschers miterleben und beobachten. Im Zuge dessen wurden auch die Dokumentationsmöglichkeiten für derartige Testszenarien vorgestellt und besprochen.

10.4.2.1 Testsetting

Die Usability Evaluierung des Prototyps 1 fand im Kindergarten D statt. Es nahmen an der Evaluierung 11 Kinder, fünf Mädchen und sechs Jungen, teil. Der Großteil davon war auch im bisherigen Child-Centered-Design-Prozess involviert. Die Kinder waren im Alter von 5 bis 6 Jahren.

Für die Durchführung der Usability-Evaluierung baute die Autorin, wie in Abbildung 38 dargestellt, in einem separaten Raum das Testsetting auf. Auf einer Spiel-Werkbank kam ein Asus eeePC mit TouchScreen als Computer zum Einsatz auf dem der Prototyp via Präsentationsmodus von Microsoft® PowerPoint® ausgeführt wurde. Zusätzlich zeichnete die Verfasserin der gegenständlichen Arbeit das Klicken und Bedienen des Prototyps durch die Kinder mittels Screenrecording-Software, in diesem Fall Morae® von TechSmith, auf. Diese spezielle Screenrecording-Software hebt Mausklicks farblich hervor, so dass diese zum einen leichter verfolgbar sind, zum anderen leichter erkennbar ist, welcher Mausklick getätigt wurde. Zusätzlich nimmt die Software den Ton des Kinds und der Testleitung auf.

Das Kind selbst nahm auf der blauen Sitzbank Platz und bediente den Computer mittels Fingertipps am Bildschirm. Die Testleitung nahm rechts neben dem Kind Platz und hatte zusätzlich Papier und Stift für Notizen zur Verfügung.

Außerdem zeichnete die Autorin das gesamte Testszenario mittels Videokamera auf um, im Falle einer defekten Screenrecording-Aufnahme, dennoch Datenmaterial zur Verfügung zu haben. Die zusätzliche Tonaufzeichnung mittels eines Audioaufnahmegeräts nahm die Verfasserin der gegenständlichen Arbeit vor, um allfällige Schwächen der im Testgerät eingebauten Mikrofone ausgleichen zu können.



Abbildung 38: Testsetting Usability Evaluierung Prototyp 1

10.4.2.2 Eingesetzte Methoden und Vorgehensweise

Die Verfasserin der gegenständlichen Arbeit führte die Usability-Evaluierung mittels apparativer, standardisierter Beobachtung durch und zeichnete die Tests komplett mittels Videokamera und mit der Software Morae® von Techsmith direkt am PC auf. Mit Morae® erfolgte das farbliche Hervorheben der Mausclicks der Kinder, was bei der Evaluierung einer TouchScreen Benutzeroberfläche unbedingt notwendig ist, da hier kein herkömmlicher Cursor angezeigt wird. Auch die Verfolgung der „Mausspur“ ist grafisch auf einem TouchScreen nicht möglich, da der Cursor nur punktuell (zum Klicken) genutzt wird und nicht mehr über der Oberfläche bewegt wird, wie bei einer herkömmlichen Maus.

Im Anschluss an den Test analysierte die Autorin das aufgenommene Videomaterial mittels standardisierter Beobachtung und wertete es aus. Die Kinder evaluierten nacheinander in Einzelsettings die Usability des Prototyps 1. Ein Test dauerte zwischen 08:05 und 12:35 Minuten, im Durchschnitt knapp 10:00 Minuten. Jeder Test begann mit dem Startbildschirm von dem aus die Kinder über Druck auf den Start-Button – die Spielfigur – zur Aufgabenauswahl kamen.

Bei der Auswertung der Ergebnisse ist zu berücksichtigen, dass die Autorin mit den Kindern keinen „klassischen Usability-Test“ durchführen konnte. Nicht gelöste Aufgabenstellungen und das „Nicht-beantworten-können“ von Fragen führen bei Kindern zu Demotivation und Verunsicherung. Daher intensivierte die Testleitung in angemessener Zeit die Hilfestellungen zu den Aufgabenstellungen und beantwortete Fragen ebenso nach angemessener Zeit selbst. Generell achtete die Verfasserin der gegenständlichen Arbeit im Testsetting darauf, dass für das Kind keine Prüfungssituation entstand.

10.4.2.3 Fragestellung

Die Usability-Evaluierung erfolgte entlang folgender halbstandardisierter Fragestellungen, die als Leitfragen zu verstehen sind. Es bestand die Möglichkeit für die Testleitung weitere Fragen, welche im Kontext geeignet erschienen, zu stellen und auch die eine oder andere Frage situativ begründet wegzulassen.

1. Wo bzw. wie kannst du eine Aufgabe auswählen?
2. Wie kann dem Schmetterling ein Schritt vorwärts gespeichert werden?
3. Wie kann der Schmetterling einen Schritt vorwärts fliegen?
4. Wie kann man einen eingegebenen Schritt löschen?
5. Wie kommst du zur Hilfe?
6. Wie kannst du eine Pause machen?

7. Wie kommst du zurück zur Aufgabenauswahl?
8. Wie kannst du das Programm beenden?
9. Bitte wähle die Aufgabe mit dem roten Herz. Was genau ist deine Aufgabe bzw. wie lautet die Aufgabenstellung?
10. Schaffst du es, dass der Schmetterling zum roten Herz fliegt?
11. Gefällt dir der Schmetterling?
12. Gefallen dir die Farben im Programm?
13. Gefällt dir, dass man die Buttons direkt antippen kann und keine Maus dafür notwendig ist?

10.4.2.4 Standardisierte Beobachtungskategorien

Aus der Fragestellung abgeleitet präzierte die Autorin Beobachtungskategorien, die helfen, sich auf die gewünschten Details zu fokussieren. Dazu entwickelte die Verfasserin der gegenständlichen Arbeit fünf Hauptkategorien, in denen sie wiederum entsprechende Unterkategorien anführte. Das Kategoriensystem mit eindeutigen Identifikationsnummern für die Usability-Evaluierung des Prototyps 1 sieht wie folgt aus:

PT101	Aufgabenauswahl und Aufgabe
PT10101	Kind kann auf Aufforderung selbständig Aufgabe auswählen
PT10102	Kind kann auf Aufforderung Aufgabe mit einer Hilfestellung auswählen
PT10103	Kind kann auf Aufforderung Aufgabe mit mehreren Hilfestellungen auswählen
PT10104	Kind nennt zu Beginn des Testsettings nur Details von der Mitte
PT10105	Kind erkennt ohne Hinweise Aufgabe im Bereich der Aufgabenstellung
PT10106	Kind kann Aufgabenstellung in eigenen Worten wiedergeben (bzw. weiß, was zu tun ist)
PT10107	Kind findet auf Anhieb Button für Aufgabenauswahl
PT102	Programmoberfläche allgemein
PT10201	Kind findet Navigationsbereich ohne Hilfestellung
PT10202	Kind findet auf Anhieb Button zum Losfahren (Start)
PT10203	Kind findet auf Anhieb Button zum Löschen
PT10204	Kind thematisiert von sich aus Zeitanzeige
PT10205	Kind thematisiert Zeitanzeige auf Nachfrage als Zeitanzeige bzw. Fortschrittsanzeige
PT10206	Kind findet Button für Beenden ohne Hilfestellung
PT10207	Kind findet Pause ohne Hilfestellung
PT10208	Kind kann Bedeutung des Doppelpfeils richtig interpretieren (WH, nochmals versuchen o.ä.)
PT10209	Kind thematisiert Befehlsliste von sich aus
PT10210	Kind drückt bei „Super!“ auf Aufgabenauswahl, um zur nächsten Aufgabe zu kommen
PT10211	Kind drückt bei „Super!“ beliebig am Bildschirm, um weiter zu kommen
PT10212	Kind drückt Stopp und ist irritiert bzw. Kind verwechselt Stopp mit Start
PT103	Hilfe
PT10301	Kind findet Button für Hilfe ohne Hilfestellung
PT10302	Kind erkennt / beschreibt, was die Hilfe anzeigt
PT10303	Kind ist durch die Anzeige der Hilfe irritiert
PT10304	Kind kann ohne Hilfestellung zur Aufgabe zurückwechseln
PT104	Aufgaben lösen
PT10401	Kind kann den Start-Button bewusst sinngemäß verwenden
PT10402	Kind kann die Richtungs-Buttons bewusst sinngemäß verwenden
PT10403	Kind kann als erste Aufgabe einen Schritt (vorwärts, seitwärts) ohne Hilfestellung fahren

PT10404	Kind kann die vom Prototyp zur Verfügung gestellte Aufgabe selbständig ohne wesentliche Hilfestellung lösen
PT10405	Kind kann die vom Prototyp zur Verfügung gestellte Aufgabe mit Hilfestellung lösen
PT10406	Kind kann die vom Prototyp zur Verfügung gestellte Aufgabe mit erheblicher Hilfestellung lösen (Lösung wird beinahe vorgegeben)
PT105	Begeisterung
PT10501	Kind möchte von sich aus weitere Aufgaben lösen
PT10502	Kind findet die Bedienung via TouchScreen gut
PT10503	Kind findet Buttongröße gut
PT10504	Kind findet Schmetterling als Spielfigur gut
PT10505	Kind findet diesen Schmetterling gut
PT10506	Kind gefallen die Farben im Arbeitsbereich (Raster, Symbole)

Tabelle 21: Kategoriensystem zur standardisierten Beobachtung der Usability-Evaluation des Prototyps 1

Die Hinweise zur Kodierung des Kategoriensystems sind in Anhang A8 zu finden.

10.4.2.5 Ergebnisse

Die Aufgabenauswahl und das Starten einer Aufgabe war eine der ersten Fragestellungen (PT10101, PT10102, PT10103), um von der Aufgabenübersicht zum Arbeitsbereich wechseln zu können. 6 der 11 Kinder, die an der Usability-Evaluierung teilnahmen, konnten auf Anhieb eine Aufgabe selbständig auswählen. Ein Kind benötigte hierbei ein wenig Hilfestellung und 3 Kinder benötigten erhebliche Hilfestellungen. Ein Kind schaffte es trotz Hilfestellungen nicht, die einzige als verfügbar angezeigte Aufgabe auszuwählen und so musste die Testleitung die Aufgabe auswählen. Ein Kind erwähnte ohne Hilfestellung, dass die Aufgabe oberhalb der Arbeitsfläche angezeigt wird (PT10105) und insgesamt 4 Kinder konnten die Aufgabenstellung in eigenen Worten formulieren (PT10106).

3 Kinder beschrieben nach Auswahl der Aufgabe am Bildschirm mit der eigentlichen Arbeitsfläche vorwiegend Details in der Mitte des Bildschirms (PT10104). Die übrigen Kinder erwähnten zumindest ein weiteres Detail außerhalb der Mitte des Bildschirms. 3 Kinder fanden auf Anhieb den Button für die Aufgabenauswahl am rechten Bildschirmrand (PT10107).

Den Navigationsbereich zur Steuerung der Spielfigur, in diesem Fall ein Schmetterling, erkannten 8 der 10 Kinder, die antworteten, auf Anhieb ohne Hilfestellung (PT10201). Den Button für Start identifizierten 3 Kinder auf Anhieb (PT10202), den Button für Löschen 7 Kinder sofort (PT10203).

6 der 9 Kinder, die danach gefragt wurden, fanden den Button für Beenden ohne weitere Hilfestellung (PT10206), dies gelang 5 der 10 Kinder für den Button für Pause ohne Hilfestellung (PT10207). 2 Kinder thematisierten von sich aus die Befehlsliste unterhalb der Arbeitsfläche (PT10209). Kein Kind thematisierte von sich aus die Zeitanzeige links oberhalb der Arbeitsfläche (PT10204), jedoch erkannten 4 Kinder auf Nachfrage, dass es sich um eine Zeitanzeige bzw. Zeitnehmung oder im weitesten Sinne um eine Fortschrittsanzeige handelt (PT10205).

Ein Kind kam in die Situation, dass ihm der Doppelpfeil (Aufforderung, die Aufgabe zu wiederholen) angezeigt wurde (PT10208) und das Kind interpretierte seine Bedeutung korrekt. Alle Kinder, die eine Aufgabe geschafft hatten, tippten beim Pop-Up „Super!“ an eine beliebige Stelle am Bildschirm, um wieder zur gewohnten Bildschirmoberfläche zurück zu wechseln (PT10211).

6 der 9 Kinder, welchen die Frage gestellt wurde, fanden auf Anhieb den Button für Hilfe (PT10301). Davon beschrieben bzw. erklärten 5 Kinder, was in der Hilfe (die ohne Text

auskommt) dargestellt wird (PT10302). Kein Kind war durch die Anzeige der Hilfe irritiert (PT10303) und 7 der 10 Kinder, die die Hilfe-Seite sahen, konnten von der Hilfe-Seite wieder selbständig zur Aufgabe zurückwechseln (PT10304).

Bis auf ein Kind konnten alle Kinder, die an der Usability-Evaluierung des Prototyps 1 teilnahmen, den Button für Start bewusst sinngemäß verwenden (PT10401). Ebenso viele Kinder konnten die Richtungs-Buttons bewusst sinngemäß verwenden (PT10402). 2 der 10 Kinder, welchen diese Aufgabe gestellt wurde, konnten auf Anhieb einen Schritt vorwärts bzw. seitwärts fahren (PT10403). 4 Kinder konnten die im Prototyp zur Verfügung stehende Aufgabe selbständig und ohne Hilfestellung lösen (PT10404), 3 Kinder benötigten Hilfestellung (PT10405) und einem Kind wurde die Lösung durch die Hilfestellung beinahe vorgegeben (PT10406). 2 Kinder konnten die Aufgabe nicht lösen.

Die Begeisterung der Kinder lässt sich wie folgt darstellen: 5 der 11 Kinder (45 %) wollten von sich aus weitere Aufgaben mit dem Programm lösen (PT10501). Die Bedienung mittels TouchScreen hoben 8 Kinder positiv hervor (PT10502), die Buttongröße empfanden 4 der 6 befragten Kinder als gut geeignet (PT10503). Den Schmetterling als Spielfigur befürworteten alle 11 an der Usability-Evaluierung teilnehmenden Kinder und 4 Kinder wollten sein Aussehen verändert wissen (PT10504, PT10505). 7 von 10 befragten Kindern gefielen die verwendeten Farben (PT10506).

10.4.2.6 Interpretation der Ergebnisse

Die Auswahl einer Aufgabe erscheint durchaus intuitiv für die Kinder zu sein, etwas mehr als die Hälfte der Kinder konnte ohne Unterstützung die verfügbare Aufgabe durch Betätigung des richtigen Buttons auswählen (PT10101). Die anderen Kinder benötigten vor allem Ermutigung dahingehend, dass sie am TouchScreen mit den Fingern tippen dürfen. Manche Kinder versuchten die deaktivierten Aufgaben, die die Autorin mittels grauen Buttons angedeutet hat, zu betätigen. Den Kindern ist noch nicht vertraut, dass deaktivierte Buttons zwar sichtbar sind, aber nicht in voller Farbe dargestellt werden.

Aus den Ergebnissen der Software-Usability-Tests in Kapitel 9.2.6 ist ersichtlich, dass die Kinder vorrangig die Augen in der Mitte des Bildschirms halten und erst auf Aufforderung die Ränder erkunden. Jedoch können nicht alle Bedienelemente und die Arbeitsfläche in der Mitte des Bildschirms angeordnet werden – es fehlt der Platz dafür. Daher legte die Verfasserin der gegenständlichen Arbeit Augenmerk darauf, die wesentlichen Bereiche farbig zu unterlegen und so voneinander abzuheben. Es scheint soweit geglückt zu sein, da nur 3 Kinder mit ihrer Wahrnehmung zu Beginn ausschließlich in der Mitte des Bildschirms blieben (PT10104).

Nur ein Kind nahm von sich aus Bezug auf die Darstellung der Aufgabenstellung im grünen Bereich links oben am Bildschirm. Die Aufgabenstellung wahrgenommen haben dürften aber insgesamt 4 Kinder, denn 3 weitere Kinder konnten diese ebenso in eigenen Worten wiedergeben (PT10105, PT10106). 3 Kinder fanden an der rechten Bildschirmseite den Button für die Aufgabenauswahl, davon hatte sich ein Kind zu Beginn nur mit den Details in der Mitte des Bildschirms beschäftigt. Mit der Zeit scheint der Blick durchaus zu wandern und eher im Moment unwichtige Buttons dürften auf der rechten Seite ganz gut aufgehoben sein.

Die Positionierung des Navigationsbereichs mit seinen großen Richtungs-Buttons auf der linken Bildschirmseite, auf Höhe der Arbeitsfläche (daraus folgt, dass die Richtungs-Buttons vertikal in der Bildschirmmitte angeordnet sind) erscheint gut gewählt. Die Mehrheit der Kinder konnte den Navigationsbereich als solchen auf Anhieb identifizieren (PT10201). Der Button für Start erscheint noch zu wenig klar, denn nur 3 Kinder fanden ihn ohne Hilfestellung (PT10202). Vermutlich irritierte hier der danebenliegende Button für Stopp, der ähnlich aussieht und von 4 Kindern genutzt wurde (PT10212). Hingegen der Button für Löschen dürfte gut geeignet sein, knapp drei Viertel der Kinder nahmen diesen ohne Hilfestellung als Möglichkeit zu löschen wahr (PT10203).

Die Buttons für Beenden (PT10206) und Pause (PT10207) fand rund die Hälfte der Kinder ohne Hilfestellung. Dies ist ein deutlich besserer Wert als beim Button für die Aufgabenauswahl, der unterhalb des Buttons Pause angesiedelt ist. Der Button Beenden ersetzt den Button Aufgabenauswahl auf dem Bildschirm Aufgabenauswahl. Nur 2 Kinder erkannten von sich aus die Befehlsliste unterhalb der Arbeitsfläche. Dieser eigentlich wichtige Bereich, er repräsentiert die eingespeicherten Befehle, scheint für die Kinder eine absolut untergeordnete Rolle zu spielen. Es gilt dies weiter zu evaluieren, denn die Aufgabenstellung war noch nicht allzu komplex.

Die Zeitnehmung (PT10204, PT10205) fiel kaum einem Kind auf und erst auf Nachfrage erwähnten 4 Kinder diese. Unter Berücksichtigung, dass die Testleitung nur bei 6 Kindern diesbezüglich nachfragte sind das immerhin zwei Drittel, die die Zeitanzeige zumindest als solche identifizierten. Bei der Usability-Evaluation spielte diese für die Kinder aber keine Rolle und sie nahmen die Zeitnehmung während des Ausprobierens nicht wahr.

Da nur ein Kind die Situation erlebte, dass der Wiederholungspfeil (PT10208) zum Vorschein kam, kann die Autorin an dieser Stelle die Gebrauchstauglichkeit desselben kaum beurteilen. Zumindest das betroffene Kind konnte den Wiederholungspfeil richtig interpretieren und wusste, wie es weitergeht. Da alle Kinder, die eine Aufgabe lösten und somit geschafft haben, das Pop-Up mit „Super!“ sahen und dann anschließend an eine beliebige Stelle klickten, um weiter machen zu können, erscheint es sinnvoll, den Kindern die Möglichkeit zu bieten, allfällige Informationen durch Klick an eine beliebige Stelle schließen zu lassen.

Die Hilfe, PT103, kam gut bei den Kindern an. Den richtigen Button identifizierten zwei Drittel der Kinder, 83 % der betreffenden Kinder konnten den Inhalt der Hilfe beschreiben bzw. wiedergeben. Kein Kind war durch die Art und Weise der Hilfe irritiert und fast alle Kinder konnten wieder selbständig zurück zur Aufgabe wechseln.

Den Kindern fiel es leicht, den Button für Start bzw. die Richtungs-Buttons sinngemäß zu verwenden (PT10401, PT10402), 90 % der Kinder konnten die Buttons korrekt identifizieren und nutzen. Mit Hilfestellungen konnten rund zwei Drittel der Kinder die im Prototyp enthaltene komplexe Aufgabenstellung lösen. Das hinter dem Prototyp liegende Konzept zum Erlernen der Programmierung einfacher und komplexer Befehlsabfolgen, scheint für die Zielgruppe zu funktionieren, wenn auch mit Unterstützung, die mit zunehmenden Komplexitätsgrad der Aufgaben ebenfalls zunimmt.

Die Kinder scheinen durchwegs begeistert von dem Prototyp und dem durch ihn repräsentierten Programm. 45 % der befragten Kinder wollten von sich aus weitere Aufgaben lösen und waren schon gespannt auf neue Aufgabenstellungen (PT10501). 72 % der Kinder gefiel die TouchScreen Bedienung (PT10502) und es zeigte sich, dass sie mit dieser kaum Probleme hatten. Kein Kind vermisste eine Computermaus. Die Buttongröße war für 66 % der befragten Kinder gut und geeignet. Der Schmetterling eroberte die Herzen der Kinder, allen gefiel ein Schmetterling als Spielfigur – nur an der Ausgestaltung hatten manche Kinder etwas auszusetzen. Die Farbgestaltung der Bildschirmoberfläche gefiel 70 % der Kinder.

Wie die informellen Ergebnisse der teilnehmenden Beobachtung der Verfasserin der gegenständlichen Arbeit an der Evaluierung des Prototyps 1 zeigen, waren alle teilnehmenden Kinder aufmerksam, zum überwiegenden Teil hoch motiviert und interessiert an der Aufgabenstellung. Die Kinder beantworteten die gestellten Fragen geduldig und die meisten Kinder hatten sichtlich Spaß an der Erkundung von Neuem. Auch die Teilnahme an einem Evaluierungs-Prozess, was für alle Kinder neu war, sorgte für Begeisterung. Die Kinder freuten sich einen Beitrag zur Entwicklung des Produkts - die grafische Benutzeroberfläche - zu leisten. Zwei Kinder waren eher schüchtern und verhielten sich zurückhaltend, waren aber dennoch sehr aufmerksam und unterstützten mit ihren Antworten.

Zusammengefasst: Der Prototyp zeigt den Weg auf, mit welchen Möglichkeiten eine für die Zielgruppe intuitive Bildschirmoberfläche einer Mikrowelt zum Programmieren lernen gestaltet werden kann. Die Kinder sind von der Gestaltung begeistert, sie finden die

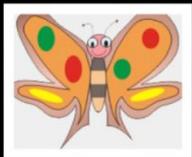
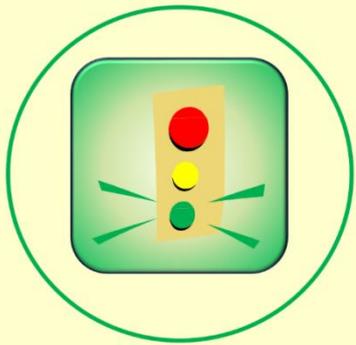
wichtigsten Buttons meist von selbst. Die Figur des Schmetterlings scheint gut gewählt, jedoch die Optik sollte noch ein wenig adaptiert werden. Es handelt sich hierbei um eine interaktive Lernumgebung, bei der die Autorin versucht, abstrakte Konzepte und Denkmuster angreifbar und leicht erlernbar zu machen, jedoch erschließt sich nicht jedem Kind die eigentliche Aufgabenstellung von allein und zur Lösung der Aufgaben benötigen die Kinder oftmals Hilfestellung. Für die nächste Version des Prototyps sollten mehrere Aufgaben in unterschiedlichen Schwierigkeitsgraden zur Verfügung stehen. So kann besser demonstriert werden, dass sich das Kind bei der Bearbeitung von Aufgabe zu Aufgabe selbst steigern und seine Fähigkeiten und Fertigkeiten in der Programmierung des Schmetterlings weiter verbessern kann.

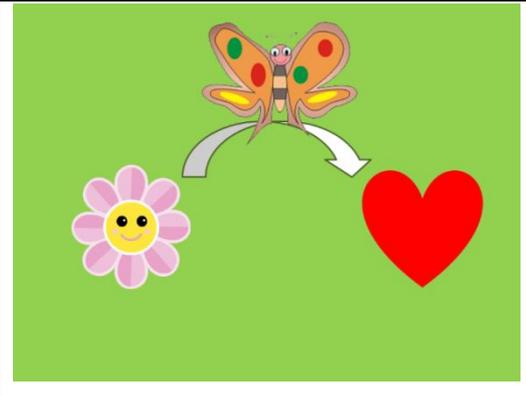
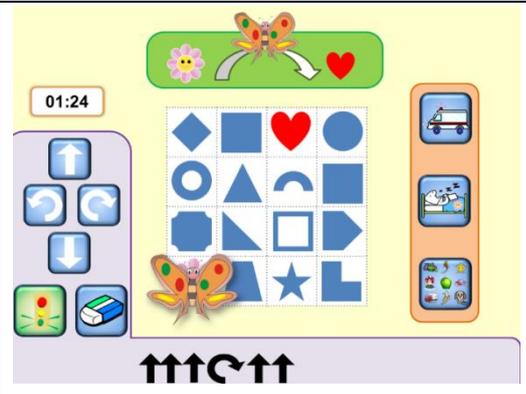
Ein Button erscheint unnötig: der Stopp-Button. Kein Kind setzte den Stopp-Button sinngemäß ein und eigentlich wurde dieser auch nicht benötigt. Daher kann dieser Button vollends weggelassen werden, was auch zu einer einfacheren Identifizierung des Start-Buttons führt, denn er wird nun dann der einzige sein, der eine Ampel abbildet.

Geschlechtsspezifische Unterschiede sind nicht explizit beobachtbar gewesen, ausgenommen von der Auswahl einer Aufgabe – hier waren die Mädchen anfangs wohl noch eher zurückhaltend, aber sowohl Mädchen als auch Jungen konnten sich auf der Bildschirmoberfläche zurechtfinden.

10.4.3 Iteration Prototyp - Version 2

Der Prototyp 2 stellt eine Evolution des Prototyps 1 dar. Daher stellt die Verfasserin der gegenständlichen Arbeit im Folgenden die Unterschiede bzw. Neuerungen basierend auf den Erkenntnissen der Usability-Evaluierung des Prototyps 1 vor. Neben einigen grafischen Überarbeitungen und der Neupositionierung von Elementen integrierte die Autorin ein grundlegend überarbeitetes Hilfesystem, das als tutorielles System agiert. Einen Überblick über den Prototyp 2 liefert die Tabelle 22.

Bild des Prototyps	Kurzbeschreibung Veränderungen
	<p>Start:</p> <p>Als eine der zentralen Neuerungen in Version 2 zeichnete die Autorin die Spielfigur, den Schmetterling, neu und gestaltete ihn kindgerechter. Auch wurden der Kopf und das Gesicht ein wenig vergrößert, so dass den Kindern die Orientierung auf der Arbeitsfläche leichter fällt.</p>
	<p>Aufgabenauswahl:</p> <p>In der Version 2 des Prototyps stehen drei Aufgaben mit unterschiedlichen Schwierigkeitsgraden zur Verfügung. Die erste Aufgabenstellung ist die, dass der Schmetterling zum gelben Kreis fliegen soll. Dazu muss er einen Schritt vorwärts fahren. Bei der Aufgabe mit dem blauen Quadrat muss der Schmetterling zwei Schritte rückwärts fahren. Bei der 3. Aufgabe zum roten Herz muss der Schmetterling die längste Strecke inkl. einer Drehung zurücklegen.</p>
	<p>Hilfe:</p> <p>Die Verfasserin der gegenständlichen Arbeit überarbeitete die Hilfe grundlegend und passte sie an die Bedürfnisse der Kinder an. Dazu wurden mit einem halbtransparenten Schwarz die Elemente „ausgeblendet“, die nicht unbedingt für die Steuerung des Schmetterlings von Nöten sind.</p> <p>Nach Aufruf der Hilfe läuft eine Animation ab, die zeigt, dass zur Lösung der im grünen Bereich angezeigten Aufgabenstellung ein Schritt vorwärts im Navigationsbereich gedrückt werden muss (angezeigt durch einen Finger, der auf den Button drückt). Danach muss der Start-Button gedrückt werden (wiederum angezeigt durch einen Finger, der auf den Button drückt) und anschließend bewegt sich der Schmetterling zum gelben Kreis, ein Feld nach vorne. Da die Aufgabe somit erfolgreich gelöst wurde, erscheint das „Super!“-Pop-up.</p>
	<p>Kontextabhängige Hilfe:</p> <p>Zusätzlich zur regulären Hilfe arbeitete die Autorin eine Hilfestellung ein, die dann erscheint, wenn ein richtiger Weg eingegeben wurde und das Kind einen anderen Button als den Start-Button betätigt. Dann soll das Symbol darauf hinweisen, den Start-Button zu betätigen um die Aufgabe abzuschließen.</p>

	<p>Kontextabhängige Hilfe:</p> <p>Nach Auswahl einer Aufgabe wird die jeweilige Aufgabenstellung bildschirmfüllend eingeblendet. Zusätzlich zeigt eine Animation des Schmetterlings, dass dieser von seinem Startpunkt, der Blume, zum roten Herz „fliegen“ bzw. fahren soll.</p> <p>Die kontextabhängige Hilfe ist ein wesentlicher Bestandteil des integrierten tutoriellen Systems, welches in der Mikrowelt enthalten sein soll.</p>
	<p>Haupt-Bildschirmoberfläche:</p> <p>Zum direkten Vergleich sind hier oben die Haupt-Bildschirmoberfläche des Prototyps 1 und darunter die Haupt-Bildschirmoberfläche des Prototyps 2 dargestellt. Die Veränderungen lassen sich wie folgt zusammenfassen:</p> <ul style="list-style-type: none"> • Die Autorin verschob die Aufgabenstellung im grünen Bereich in die Mitte des Bildschirms, damit die Kinder diesen leichter finden. • Die Schaltflächen wurden noch etwas größer, damit die Bedienung leichter fällt und kleinere Symbole leichter erkennbar sind. • Den rechten, orangen Bereich mit den kontextabhängig zur Verfügung stehenden Buttons verschob die Verfasserin der gegenständlichen Arbeit ebenfalls etwas mehr in Richtung Mitte, damit er leichter gefunden werden kann. • Die Arbeitsfläche wurde von der Autorin weiß unterlegt, damit sie klarer erkennbar ist. • Der Stopp-Button wurde entfernt. • Den Start-Button hinterlegte die Verfasserin der gegenständlichen Arbeit grün, damit er einfacher aufgefunden werden kann und selbsterklärender wird. • Unterhalb des Schmetterlings befindet sich nun auch ein Symbol, das Startsymbol. Hier ist es eine Blume. • Die Autorin glich alle Elemente in ihrer Position und Größe an die eben aufgelisteten Veränderungen an.
	<p>Bildschirmoberfläche während der Bearbeitung:</p> <p>In diesem Fall wurden die Befehle eingegeben, die benötigt werden, dass der Schmetterling die Aufgabe (Bewegung vom Startsymbol, der Blume zum Herz) lösen kann. Es müsste jetzt nur mehr der Start-Button betätigt werden und schon „fliegt“ der Schmetterling zum roten Herz. Die eingegebenen Befehle sind in der Befehlsliste unterhalb des Arbeitsbereichs ersichtlich.</p>

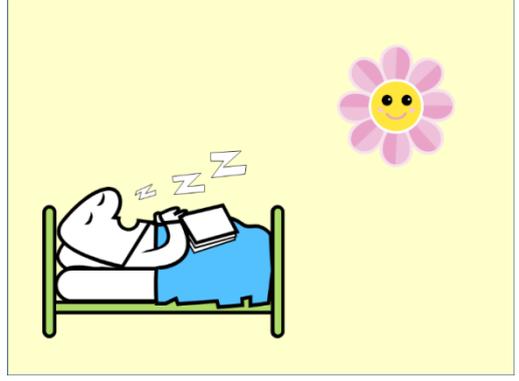
	<p>Aufgabe gelöst:</p> <p>An dieser Bildschirmoberfläche hat die Autorin nichts geändert, außer das Grundlayout, wie im gesamten Prototyp.</p>
	<p>Pause:</p> <p>Die Verfasserin der gegenständlichen Arbeit positionierte das Pause-Bild auf gelbem Hintergrund, um es weniger bedrohlich wirken zu lassen und ein Abheben von anderen Bildschirmen zu unterstützen. So wirkt die Pause wesentlich freundlicher.</p>
	<p>Aufgabe nicht geschafft:</p> <p>Ähnlich zur Pause legte die Autorin das Symbol auf den freundlicheren gelben Hintergrund, damit das Symbol nicht zu bedrohlich wirkt.</p>

Tabelle 22: Übersicht Prototyp 2

10.4.4 Evaluation Prototyp Version 2

In der Session 5 des Child-Centered-Design-Prozesses evaluierten die teilnehmenden Kinder im Kindergarten D die Usability des Prototyps 2. Zur Evaluation des Prototyps stellte die Verfasserin der gegenständlichen Arbeit die gewünschte Demo-Funktionalität, bei der der Schmetterling in drei verschiedenen Schwierigkeitsgraden von seinem Ausgangspunkt zur jeweiligen geometrischen Figur fliegen soll, in Form einer Abfolge von einzelnen Präsentationsfolien her. Die finale Präsentation und somit der Prototyp umfasst 87 Präsentationsfolien und deckt die Varianten zur Lösung der Aufgaben ab, die die Kinder mit den kürzest möglichen Wegen finden können.

Zu Beginn des Testszenarios erklärte die Autorin dem jeweiligen Kind das Zustandekommen der nun basierend auf den Testergebnissen der Evaluierung des Prototyps in Version 1 vorliegenden Iteration des Prototyps in Version 2. Die Verfasserin der gegenständlichen Arbeit wiederholte gemeinsam mit dem Kind die schon bei der Evaluierung des Prototyps in Version 1 besprochenen Ziele der Evaluierung, die vom Kind einzunehmende Rolle und die generelle Vorgehensweise bei der Evaluierung. Auch die Unterschiede zwischen einem klickbaren Prototyp, der simuliert wird und bei dem noch nicht alle Funktionalitäten bzw.

Aufgaben zur Verfügung stehen und einem tatsächlich funktionalen Software Produkt waren nochmals Bestandteil der Einleitung des Testszenarios.

Die Kinder können durch die Teilnahme an der Evaluierung des Prototyps in Version 2 die Weiterentwicklung von Produkten beobachten und sehen, wie ihr Feedback zu Änderungen am Produkt beiträgt. Die vier Kinder im Alter von 3 bis 4 Jahren, die in dieser Session zur Durchführung des adaptierten Child-Centered-Design-Prozesses hinzukamen, können vorwiegend in der Rolle als Benutzerinnen und Benutzer aber auch als Testerinnen und Tester erste Erfahrungen mit der experimentellen Evaluierung von Prototypen sammeln.

10.4.4.1 Testsetting

Das Testsetting unterscheidet sich nicht von dem der Usability-Evaluation des Prototyps 1 (siehe Kapitel 10.4.2). Sowohl Ort als auch Art der Usability-Evaluation blieben gleich.

Zu Beginn führte die Autorin die Usability-Evaluation mit den 5- bis 6-jährigen Kindern durch, die auch an der Usability-Evaluation des Prototyps 1 teilgenommen hatten. Anschließend folgte die Durchführung der Usability Evaluierung mit vier Kindern im Alter von 3 bis 4 Jahren, um zu testen, ob der Prototyp für diese Altersgruppe ebenso geeignet ist.

10.4.4.2 Eingesetzte Methoden und Vorgehensweise

Die zur Usability-Evaluierung des Prototyps 2 eingesetzten Methoden und die Vorgehensweise unterscheiden sich nicht von der Usability-Evaluierung des Prototyps 1 (siehe Kapitel 10.4.2.2).

10.4.4.3 Fragestellung

Die Fragestellung bzw. die Leitfragen für die Durchführung der Usability-Evaluierung blieben bis auf Frage 9 und Frage 10 unverändert zum Prototyp 1 (siehe Kapitel 10.4.2.3) Diese Fragen passte die Verfasserin der gegenständlichen Arbeit an die neuen Gegebenheiten, drei verschiedene Aufgaben, an.

10.4.4.4 Standardisierte Beobachtungskategorien

Basierend auf den Kategorien für die Usability-Evaluierung des Prototyps 1 adaptierte die Autorin die standardisierten Beobachtungskategorien für die Usability-Evaluierung des Prototyps 2.

- Die Kategorie PT10105 adaptierte die Autorin für den Prototyp 2 und ergänzte diese um „Aufgabenanimation“, da die Kinder nun nach Aufruf einer Aufgabe eine kurze Animation angezeigt bekommen, die die Aufgabe erklärt.
- Die Kategorie PT10212 wurde gestrichen, da es beim Prototyp 2 keinen Stopp-Button mehr gibt.
- Die Kategorien PT10204 und PT10205 strich die Verfasserin der gegenständlichen Arbeit, da die Zeitanzeige im neuen Prototyp im Vergleich zum vorhergegangenen Prototyp nicht verändert wurde.
- Die Kategorie PT10303 ergänzte die Autorin in der Kategorie PT20303 um den Zusatz „(Animation)“.
- In der Hauptkategorie PT104 nahm die Verfasserin der gegenständlichen Arbeit für die Hauptkategorie PT204 umfangreiche Änderungen vor. So wurde die Kategorie PT10403 ersatzlos gestrichen, da diese einfache Aufgabe nun in der Aufgabe 1 enthalten ist (PT20403, PT20406, PT20409)
- Für die Aufgaben 2 und 3 nahm die Autorin entsprechende Kategorien in die Hauptkategorie PT204 mit auf.
- In der Hauptkategorie PT105, folglich die Hauptkategorie PT205, wurde die Kategorie PT10502 gestrichen – da die Bedienung via TouchScreen bereits beim ersten Prototyp evaluiert wurde.

- In die Hauptkategorie PT205 fügte die Autorin die Kategorie PT20505 hinzu um festzustellen, ob Kinder Unterschiede zwischen den beiden Prototypen thematisieren.
- Die Kategorien PT20506, PT20507 und PT20508 fügte die Verfasserin der gegenständlichen Arbeit explizit für die 3- und 4-jährigen Teilnehmerinnen und Teilnehmer an der Usability-Evaluierung ein.

Das gesamte Kategoriensystem sieht daher wie folgt aus (siehe Tabelle 23):

PT201	Aufgabenauswahl und Aufgabe
PT20101	Kind kann auf Aufforderung selbständig Aufgabe auswählen
PT20102	Kind kann auf Aufforderung Aufgabe mit einer Hilfestellung auswählen
PT20103	Kind kann auf Aufforderung Aufgabe mit mehreren Hilfestellungen auswählen
PT20104	Kind nennt zu Beginn des Testsettings nur Details von der Mitte
PT20105	Kind kann nach erster Aufgabenanimation selbständig zum Arbeitsbereich wechseln
PT20106	Kind kann Aufgabenstellung in eigenen Worten wiedergeben (bzw. weiß, was zu tun ist)
PT20107	Kind findet auf Anhieb Button für Aufgabenauswahl
PT202	Programmoberfläche allgemein
PT20201	Kind findet Navigationsbereich ohne Hilfestellung
PT20202	Kind findet auf Anhieb Button zum Losfahren (Start)
PT20203	Kind findet auf Anhieb Button zum Löschen
PT20204	Kind findet Button für Beenden ohne Hilfestellung
PT20205	Kind findet Pause ohne Hilfestellung
PT20206	Kind kann Bedeutung vom Doppelpfeil richtig interpretieren (WH, nochmals versuchen o.ä.)
PT20207	Kind thematisiert Befehlsbereich von sich aus
PT20208	Kind drückt bei „Super!“ auf Aufgabenauswahl, um zur nächsten Aufgabe zu kommen
PT20209	Kind drückt bei „Super!“ beliebig am Bildschirm, um weiter zu kommen
PT203	Hilfe
PT20301	Kind findet Button für Hilfe ohne Hilfestellung
PT20302	Kind erkennt / beschreibt, was die Hilfe anzeigt
PT20303	Kind ist durch die Art der Anzeige der Hilfe irritiert (Animation)
PT20304	Kind kann ohne Hilfestellung zur Aufgabe zurückwechseln
PT20305	Kind findet auf Nachfrage / kurze Erklärung die Art der Anzeige der Hilfe gut
PT204	Aufgaben lösen
PT20401	Kind kann den Start-Button bewusst sinngemäß verwenden
PT20402	Kind kann die Richtungs-Buttons bewusst sinngemäß verwenden
PT20403	Kind kann Aufgabe 1 selbständig ohne wesentliche Hilfestellung lösen
PT20404	Kind kann Aufgabe 2 selbständig ohne wesentliche Hilfestellung lösen
PT20405	Kind kann Aufgabe 3 selbständig ohne wesentliche Hilfestellung lösen
PT20406	Kind kann Aufgabe 1 mit Hilfestellung lösen
PT20407	Kind kann Aufgabe 2 mit Hilfestellung lösen
PT20408	Kind kann Aufgabe 3 mit Hilfestellung lösen
PT20409	Kind kann Aufgabe 1 mit erheblicher Hilfestellung lösen (Lösung wird beinahe vorgegeben)
PT20410	Kind kann Aufgabe 2 mit erheblicher Hilfestellung lösen (Lösung wird beinahe vorgegeben)
PT20411	Kind kann Aufgabe 3 mit erheblicher Hilfestellung lösen (Lösung wird beinahe vorgegeben)
PT205	Begeisterung
PT20501	Kind möchte von sich aus weitere Aufgaben lösen

PT20502	Kind findet Buttongröße gut
PT20503	Kind findet den Schmetterling von Prototyp 2 besser
PT20504	Kind gefallen die Farben im Arbeitsbereich (Raster, Symbole)
PT20505	Kind erkennt / thematisiert Unterschiede zum Prototyp 1
PT20506	Kind findet das Programm schön
PT20507	Kind würde mit dem Programm spielen
PT20508	Kind findet, man kann damit allein spielen

Tabelle 23: Kategoriensystem zur standardisierten Beobachtung der Usability-Evaluation des Prototyps 2

Die Hinweise zur Kodierung des Kategoriensystems sind in Anhang A9 zu finden.

10.4.4.5 Ergebnisse

Bei der Ergebnisdarstellung ist zu berücksichtigen, dass, bis auf die 3- und 4-jährigen Teilnehmerinnen und Teilnehmer der Usability Evaluierung des Prototyps 2, die Kinder, die den Prototyp 2 evaluieren, auch an der Usability-Evaluierung des Prototyps 1 teilgenommen haben und daher schon über eine gewisse Vorerfahrung verfügen. Auch ein Lerneffekt wird sich bei den Kindern eingestellt haben, so dass ihnen die Bedienung des Prototyps leichter fallen wird – wurde ja die grundlegende Bedienung in Version 2 nicht verändert. Darüber hinaus folgt der Ergebnis-Darstellung eine gesonderte Darstellung der Ergebnisse der 3- bis 4-jährigen Kinder.

10 der 11 Kinder im Alter von 5 bis 6 Jahren konnten auf Aufforderung selbständig zu einer Aufgabe wechseln (PT20101), ein Kind benötigte dazu ein wenig Hilfestellung (PT20102). 2 Kinder nannten zu Beginn der Usability-Evaluierung nur Details auf der Mitte des Bildschirms (PT20104). 8 Kinder (73 %) konnten, nach der Anzeige der Animation zur Erklärung der Aufgabenstellung, selbständig zur Aufgabe wechseln (PT20105), davon konnten 6 Kinder die Aufgabenstellung in eigenen Worten erklären (PT20106), was auch ein weiteres Kind konnte. 5 Kinder fanden den Button für die Aufgabenauswahl auf Anhieb (PT20107).

82 % der 5- bis 6-jährigen Kinder fanden den Navigationsbereich auf Anhieb ohne Hilfestellung (PT20201), 55 % den Start-Button (PT20202), 87 % der Kinder den Button für Löschen (PT20203) und rund zwei Drittel den Button für Beenden (PT20204). 7 der 9 befragten Kinder fanden den Button für Pause ohne Hilfestellung (PT20205). 3 Kinder kamen in die Situation, dass ihnen der Doppelpfeil zur Wiederholung einer Aufgabe angezeigt wurde – 2 Kinder konnten den Doppelpfeil korrekt interpretieren (PT20206). Ein Kind thematisierte von sich aus den Befehlsbereich (PT20207) und 91 % der Kinder schafften es sofort, bei Anzeige von „Super!“ für das erfolgreiche Lösen einer Aufgabe, selbständig zu nächsten Aufgabe zu wechseln (PT20208, PT20209).

Alle Kinder im Alter von 5 bis 6 Jahren, die die Testleitung dazu befragte, fanden auf Anhieb den Button für die Hilfe (PT20301). 7 der 11 Kinder konnten in eigenen Worten beschreiben, was die Hilfe darstellt (PT20302). Von den 7 Kindern waren 2 Kinder ob der Animation irritiert und 2 weitere Kinder ebenso (PT20303). 90 % der Kinder, die an der Usability-Evaluierung des Prototyps 2 teilnahmen, konnten selbständig ohne Hilfestellung von der Hilfe zurück zur eigentlichen Aufgabe wechseln (PT20304). 5 der 7 befragten Kinder empfanden auf Nachfrage die Hilfe als hilfreich und gut dargestellt (PT20305).

10 der 11 5- bis 6-jährigen Kinder (91 %) konnten den Start-Button bewusst sinngemäß verwenden (PT20401) und 8 der 11 Kinder (73 %) konnten die Richtungs-Buttons bewusst sinngemäß verwenden (PT20402). Die neuen Aufgaben 1 und 2 (Aufgabe 3 entspricht annähernd der Aufgabe aus Prototyp 1) konnten von den Kindern mit unterschiedlichen Hilfestellungen gelöst werden. Aufgabe 1 konnten 6 Kinder selbständig ohne Hilfestellung (PT20403), 3 Kinder mit Hilfestellung (PT20406) und ein Kind mit erheblicher Hilfestellung (PT20409) lösen. Aufgabe 2 lösten 4 Kinder selbständig ohne Hilfestellung (PT20404), 3

Kinder mit Hilfestellung (PT20407) und 4 Kinder mit erheblicher Hilfestellung (PT20410). Die dritte Aufgabe konnte von 3 Kindern der 8 Kinder, die die Aufgabe gestellt bekamen, selbständig ohne Hilfestellung (PT20405), von 4 Kindern mit Hilfestellung (PT20408) und von einem Kind nicht gelöst werden.

Mehr als die Hälfte der Kinder im Alter von 5 bis 6 Jahren, die an der Usability-Evaluierung des Prototyps 2 teilnahmen, wollten von sich aus weitere Aufgaben lösen (PT20501). 2 Kinder befragte die Testleitung bezüglich der Buttongröße und beide befanden diese für gut (PT20502). Der Schmetterling des Prototyps 2 gefiel 6 Kindern der 11 Kinder besser als der des Prototyps 1 (PT20503). 87 % der befragten 5- bis 6-jährigen Kinder sind der Meinung, dass die Farben gut gewählt wurden und ihnen Gefallen bereiten (PT20504). 82 % der Kinder thematisierten Unterschiede zwischen dem Prototyp 1 und 2 (PT20505).

Ein Kind der Gruppe der 3- bis 4-jährigen Kinder konnte selbständig eine Aufgabe auswählen (PT20101), mit Hilfestellung schafften dies 2 weitere Kinder (PT20102). Alle anderen Kategorien in der Hauptkategorie PT201 konnten nur negativ oder gar nicht kodiert werden, da entweder die Kinder die notwendigen Aktionen nicht ausführen konnten, die notwendigen Antworten nicht zur Verfügung standen oder die Testleitung die entsprechenden Fragen nicht stellte um die Kinder nicht zu demotivieren.

In der Hauptkategorie PT202 konnte ein Kind der 3- bis 4-Jährigen sowohl den Button für das Löschen finden (PT20203) als auch bei „Super!“ mit Klick auf die Aufgabenauswahl wieder zur nächsten Aufgabe wechseln.

2 der Kinder im Alter von 3 bis 4 Jahren konnten den Start-Button sinngemäß verwenden (PT20401) und die erste Aufgabe mit erheblicher Hilfestellung lösen (PT20409). Eines der beiden Kinder konnte die zweite Aufgabe mit erheblicher Hilfestellung lösen (PT20410) und eines die dritte (PT20411).

Von sich aus weitere Aufgaben lösen (PT20501) wollten 2 der 3- bis 4-jährigen Kinder. Diese beiden Kinder fanden die Farben der Benutzeroberfläche ansprechend (PT20504), eines der beiden äußerte sich positiv zur Buttongröße (PT20502). 3 Kinder fanden das Programm schön (PT20506), davon würden 2 Kinder mit dem Programm spielen wollen (PT20507). Von diesen beiden Kindern war eines der Meinung, dass das Programm so gestaltet ist, dass es damit allein, ohne weitere Hilfe, spielen kann (PT20508).

10.4.4.6 Interpretation der Ergebnisse

Die 5- bis 6-jährigen Kinder konnten gegenüber dem Prototyp 1 den Prototyp 2 deutlich selbständiger bedienen. Die Kategorie PT201 zeigt, dass die Kinder selbständig die Aufgaben auswählen konnten (PT20101), nach der angezeigten Animation zur eigentlichen Aufgabe wechseln konnten (PT20105) und rund die Hälfte der Kinder fand den Button für die Aufgabenauswahl auf Anhieb (PT20107).

Auch bei der allgemeinen Bedienung der Programmoberfläche (PT202) zeigt sich, dass diese den Kindern im Alter von 5 bis 6 Jahren leichter von der Hand ging, als noch beim ersten Prototyp. Natürlich ist ein gewisser Lerneffekt zu bemerken, trotzdem fällt es den Kindern leichter, die entsprechenden Buttons, wie Start (PT20202) oder Pause (PT20205) oder Löschen (PT20203) zu finden und zu betätigen. Aber auch beim Prototyp 2 erwähnt kaum ein Kind den Befehlsbereich von sich aus (PT20207), was dafür spricht, dass ihn die Kinder entweder nicht wahrnehmen, weil er ganz unten am Bildschirmrand angesiedelt ist oder dass sie mit den darin dargestellten Informationen, den eingegebenen Richtungsbefehlen, wenig bis gar nichts anfangen können.

Wie auch beim Prototyp 1 kam die Hilfe bei den 5- bis 6-jährigen Kindern sehr gut an, alle Kinder konnten sie aufrufen (PT20301), von der Hilfe wieder zur Aufgabe zurückwechseln (PT20304) und die Mehrheit konnte darlegen, was in der Hilfe dargestellt wird (PT20302). Die Animation der Hilfe, die in ihrer Intention als Teil eines tutoriellen Systems wirken soll, rief zu Beginn bei manchen Kindern Irritation hervor. Denn manche Kinder waren

verwundert, dass der Schmetterling einen Schritt vorwärts fuhr, obwohl sie nichts dergleichen eingegeben hatten (PT20303). Auf Erklärung und Nachfragen hin gaben die Kinder jedoch an, dass die Darstellung hilfreich ist (PT20305).

Die Bedienung der Programmoberfläche scheint den Kindern im Alter von 5 bis 6 Jahren eher leicht von der Hand zu gehen, auch wenn beim Prototyp 1 91 % der Kinder die Richtungs-Buttons bewusst sinngemäß einsetzen konnten (PT10402) und beim Prototyp 2 nur noch 73 %, was sich möglicherweise aus der Anzahl der zu lösenden Aufgaben, eine beim Prototyp 1 und drei beim Prototyp 2, ergibt. Die unterschiedlichen Schwierigkeitsgrade äußerten sich bei jedem Kind etwas anders – nur ein Kind konnte alle drei Aufgaben ohne Hilfestellungen lösen (PT20403, PT20406, PT20409), 2 Kinder konnten zumindest die ersten beiden Aufgaben ohne Hilfestellungen (PT20403, PT20406) lösen.

Die Begeisterung der 5- bis 6-jährigen Kinder ist mit dem Prototyp 2 noch etwas mehr gestiegen. Knapp 55 % der Kinder wollten von sich aus weitere Aufgaben lösen (PT20501), wohingegen knapp 45 % der Kinder mit dem Prototyp 1 noch weitere Aufgaben lösen wollten (PT10501). Die Buttongröße ist nach wie vor in Ordnung – die leichte Vergrößerung scheint der allgemeinen Bedienbarkeit gut getan zu haben (PT20502, PT10503). Die Meinungen der Kinder zum Aussehen des Schmetterlings sind bezüglich der unterschiedlichen Varianten der Spielfigur halbwegs ausgeglichen – beide Schmetterlinge der Prototypen 1 und 2 kamen gut an. Tendenziell kam der selbstgezeichnete Schmetterling in Prototyp 2 (PT20503) eine Spur besser bei den Kindern an, als das fotorealistische Abbild in Prototyp 1 (PT10505). Die Farbgestaltung im Prototyp 2 kam besser an als die des Prototyps 1. Knapp 88 % der 5- bis 6-jährigen Kinder gefällt die Farbauswahl (PT20504). 9 der 11 Kinder thematisierten Unterschiede zum Prototyp 1 (PT20505), was zeigt, dass sich die Kinder durchaus an die Details des Prototyps 1 erinnern und diese Details nun auch in Relation zum Prototyp 2 setzen konnten.

Die Autorin beobachtete während der Usability-Evaluation des Prototyps 2 keine (generalisierbaren) geschlechtsspezifischen Unterschiede.

Die Gruppe der 3- bis 4-jährigen Kinder muss separat betrachtet werden: 3- bis 4-jährige Kinder können mit einer Mikrowelt, die durch den Prototyp 2 simuliert wurde, arbeiten, allerdings nur unter Anleitung und Beisein einer Tutorin oder eines Tutors. Diese Person muss entsprechende Hilfestellung geben und für Fragen zur Verfügung stehen. Den jüngsten Kindern hat das Programm gut gefallen und 2 von ihnen konnten die Aufgaben mit entsprechender Hilfestellung lösen. Ein Kind hatte extreme Scheu davor, den TouchScreen zu berühren und hat daher nicht selbständig versucht die Aufgaben zu lösen, die Testleitung musste für das Kind die jeweiligen Buttons bedienen.

Über geschlechtsspezifische Unterschiede bei den jüngsten Kindern können ob der zu geringen Grundgesamtheit von der Autorin keine Aussagen getroffen werden.

Aus ihrer teilnehmenden Beobachtung kann die Verfasserin der gegenständlichen Arbeit folgende informelle Erkenntnisse ableiten: Wie auch schon bei der Evaluierung des ersten Prototyps waren die Kinder bei der Evaluierung des Prototyps 2 sehr engagiert und motiviert. Die jüngsten Kinder brauchten einige Hilfestellungen, um am Prozess der Evaluierung teilzunehmen - so musste die Testleitung bei der Bedienung des Prototyps assistieren oder die Kinder mehrmals zu Aussagen ermutigen. Je jünger die Kinder, desto eher sind diese schüchtern und womöglich auch verschlossen. Die älteren der teilnehmenden Kinder wollten von sich aus noch mit dem Prototyp 2 experimentieren und mehr Aufgaben lösen, als der Prototyp zur Verfügung stellte. Die Kinder waren derart begeistert, dass sie des Öfteren nachfragten „Wann kommt ihr denn wieder?“. Alle Kinder waren sehr gerne Teil der Evaluierung und berichteten stets voller Stolz vor dem Raum am Gang den Kindergartenpädagoginnen und -pädagogen sowie ihren Eltern, was sie alles Spannendes erlebt haben. Die Kinder leisteten somit nicht nur einen wertvollen Beitrag zur Ausgestaltung einer Benutzeroberfläche, sondern sie waren von sich aus hoch motivierte Teilnehmerinnen und Teilnehmer am gesamten Prozess und zeigten auch dementsprechend ihr Interesse.

Zusammengefasst: Den Kindern hat die Möglichkeit aus mehreren Aufgaben wählen zu können gefallen und sie waren durchwegs mit Begeisterung und Interesse bei der Sache. Die größten Usability-Schwächen aus Prototyp 1 konnten beseitigt werden und die Kinder kamen mit der überarbeiteten Benutzeroberfläche sehr gut zurecht. Je jünger die Kinder sind, desto eher brauchen sie eine Einschulung und Tutorium bzw. zu Beginn auch eine Begleitperson, die gemeinsam mit dem Kind die Mikrowelt erkundet und die wichtigsten Funktionen erklärt.

Das integrierte und prototypisch umgesetzte tutorielle System kann für die Kinder hilfreich sein, es muss jedoch sehr wohlbedacht gestaltet werden. Die Animation der Aufgabenstellung war den meisten Kindern eine gute Unterstützung. Auch die Animation der Hilfe war durchaus hilfreich, nur einige Kinder waren beim ersten Sehen etwas irritiert. Sie waren verwundert, dass der Schmetterling fuhr, ohne dass sie entsprechende Befehle eingegeben haben. Es bedurfte Erklärungen, damit sie nachvollziehen konnten, warum hier eine Animation läuft und sich der Schmetterling selbständig bewegte. Nach diesen Erklärungen empfanden sie die Hilfe als gut und waren mitunter sogar begeistert.

10.5 Beitrag des adaptierten Child-Centered-Designs zur frühen Technikbildung in Informatik

Korrespondierend zu den Lernergebnissen des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ (siehe Kapitel 8.2) tragen die Aktivitäten des adaptierten Child-Centered-Designs, welches in der gegenständlichen Arbeit um Cooperative-Inquiry und Bonded-Design erweitert wurde (siehe Kapitel 5.2.3.3 und die Einleitung zu Kapitel 10), in unterschiedlichen Bereichen zur Förderung der 3- bis 6-jährigen Kinder bei. Es zeigte sich bei der Durchführung des Child-Centered-Designs, dass die Erweiterung des klassischen Child-Centered-Designs um Cooperative-Inquiry und Bonded-Design mit 5- bis 6-jährigen Kindern im Kindergarten möglich ist. 3- bis 4-jährige Kinder wurden als Benutzerinnen und Benutzer sowie Testerinnen und Tester in den Prozess eingebunden. Wobei die 5- bis 6-jährigen Kinder im Sinne des Cooperative-Inquiry zusätzlich als Gestaltungspartnerinnen und Gestaltungspartner sowie als Informantinnen und Informanten agierten (gemäß Druin's-Modell, siehe Kapitel 5.2.1). Weiters sind die Kinder in der Rolle Lernende aktiv am Prozess beteiligt und sind so in der Lage, Lernergebnisse und Kompetenzen zu erwerben (siehe Tabelle 24).

Der adaptierte Child-Centered-Design-Prozess umfasst vier Phasen (siehe Kapitel 10.1). In der ersten Phase, Symbole zeichnen (siehe Kapitel 10.2), üben die Kinder offenen Aufgaben- und Fragestellungen zu folgen bzw. diese zu beantworten (vgl. LE3.4), kooperativ Aufgaben zu bearbeiten (vgl. LE2.13), Symbole zu informatischen Begriffen zu gestalten (vgl. LE3.1, LE3.3, LE2.2 und LE1.5) und selbsterstellte Artefakte der Gruppe zu präsentieren (vgl. LE3.2). Darüber hinaus rufen die Kinder (früher) erworbenes themenrelevantes Wissen ab und wenden es zur Lösung neuer Aufgabenstellungen an (vgl. LE1.7). Fragenentwickelnde Begleitung der Kinder beim Zeichnen der Symbole ermöglicht den Kindern, auf Hilfestellungen zu reagieren und diese auch zu erleben oder gar durch Gegenfragen zu motivieren (vgl. LE2.14).

Lernergebnisse	Phase 1: Zeichnen	Phase 2: Card Sorting	Phase 3: Prototyp 1 testen	Phase 4: Prototyp 2 testen
LE1.1				
LE1.2				
LE1.3				
LE1.4				
LE1.5	x			
LE1.6				
LE1.7	x		x	x

Lernergebnisse	Phase 1: Zeichnen	Phase 2: Card Sorting	Phase 3: Prototyp 1 testen	Phase 4: Prototyp 2 testen
LE2.1			x	x
LE2.2	x	x	x	x
LE2.3			x	x
LE2.4			x	x
LE2.5				
LE2.6				
LE2.7				
LE2.8				
LE2.9				
LE2.10				
LE2.11			x	x
LE2.12			x	x
LE2.13	x			
LE2.14	x		x	x
LE3.1	x		x	x
LE3.2	x			
LE3.3	x			
LE3.4	x	x	x	x
LE3.5		x	x	x

Tabelle 24: Lernergebnisse Child-Centered-Design

Beim Card-Sorting (siehe Kapitel 10.3), Phase zwei des adaptierten Child-Centered-Design-Ansatzes, trainieren die teilnehmenden Kinder ihr eigenes Handeln zu kommentieren (vgl. LE3.5), offene Frage- und Aufgabenstellungen zu beantworten bzw. diesen zu folgen (vgl. LE3.4) und Symbole zu informatischen Fachbegriffen zuzuordnen (vgl. LE2.2). Die Kinder interagieren mit der Testleitung, treffen Entscheidungen und arbeiten über einen längeren Zeitraum an einer Aufgabe. Weiter üben die Kinder, durch die Aufgabenstellung pro Bedeutungskategorie zu beurteilen, welches Symbol ihnen am besten gefällt, Präferenzen anzugeben.

Die Phasen 3 und 4 (siehe Kapitel 10.4), die Evaluierungen eines Prototyps und seiner Iterationen erlauben den teilnehmenden Kindern eine vielseitige Auseinandersetzung mit informatischen Themen. Es ergibt sich für sie die Möglichkeit, sich in einem ihnen unbekanntem Prototyp zu orientieren und diesen mittels „trial and error“ zu erkunden (vgl. LE2.3), Objekte via Touchscreen auszuwählen, einen Befehl auf der Benutzeroberfläche des Prototyps auszulösen sowie einen Zusammenhang zwischen den Befehlen, die sie mittels Touchscreen auslösen, und den Auswirkungen im Programm herzustellen (vgl. LE2.4).

Das Test-Setting erlaubt den Kindern ihr eigenes Handeln zu kommentieren (vgl. LE3.5), mit der Testleitung zu interagieren, Talk Aloud zu üben und in eigenen Worten zu beschreiben, was sie auf der Benutzeroberfläche sehen. Daraus folgend können Kinder hier erste Schritte in Richtung Thinking Aloud gehen und versuchen ihre Gedanken, wie etwas zu bedienen sein könnte, zu verbalisieren. Auch werden Kinder im Rahmen der Evaluation ermutigt, für sie Irritierendes (z.B. missverständliche Symbole, Farben etc.) zu nennen.

Die Kinder trainieren sich über einen längeren Zeitraum mit einer Aufgabenstellung zu beschäftigen, themenrelevantes Vorwissen abzurufen und wiederzugeben (vgl. LE1.7) sowie informatische Begriffe in eigene Worte zu fassen (vgl. LE2.2). Auch hier, bei der Evaluierung des Prototyps, üben die Kinder offene Frage- und Aufgabenstellungen zu beantworten bzw. ihnen zu folgen (vgl. LE3.4), einfache und komplexe Programmieraufgaben zu lösen (vgl. LE2.1), sich im Raum zu orientieren und Richtungen zu benennen (vgl. LE2.11) und die Perspektive eines Avatars (einer Spielfigur) einzunehmen (vgl. LE2.12). Hinsichtlich der

Gestaltung eines Avatars werden die Kinder animiert Präferenzen zu Farbe und Aussehen zu äußern, sowie in Bezug auf die Mikrowelt zur gewünschten Größe von Buttons, farbigen Gestaltung und Anordnung der Symbole Stellung zu nehmen. Durch die Iteration in der Entwicklung des Prototyps ergibt sich für die Kinder die Möglichkeit, Unterschiede zwischen den Versionen zu erkennen, zu benennen und die Versionen hinsichtlich ihrer jeweiligen Bedienung zu vergleichen.

Darüber hinaus erleben die teilnehmenden Kinder die Nutzung eines integrierten tutoriellen Systems und dessen Hilfestellungen, um diese zur Problemlösung selbständig anwenden zu können (vgl. LE2.1, LE2.3 und LE2.14). Im Zuge der Evaluierung der Prototypen werden die Kinder wiederholt ermutigt, Unterschiede zwischen einem Prototyp eines Programms und einem fertigen, finalen Programm selbst zu erkennen und in eigenen Worten anzusprechen.

Die genannten Lernergebnisse unterstützen den Kompetenzerwerb, wie in Kapitel 8.2.4 dargelegt. Eine Teilnahme am Child-Centered-Design-Prozess als Maßnahme zur frühen Technikbildung fördert somit die fachlichen Kompetenzen Informatik, Mathematik und Technik allgemein, sowie die akademischen Kompetenzen Gestalten, Dokumentieren, Reflektieren, Kooperativ Arbeiten, Kreativ Denken, Entscheidungen treffen, Probleme lösen, Kommunizieren, Präsentieren und Räumlich Denken.

11 Konsolidierung der Ergebnisse und Fazit

Dieses Kapitel stellt den Abschluss der gegenständlichen Forschungsarbeit dar. In Kapitel 11.1 fasst die Autorin die Resultate der gegenständlichen Arbeit in einer Übersicht zusammen und referenziert auf die korrespondierenden Inhalte dieser Forschungsarbeit. Die Hypothesenüberprüfung führt die Verfasserin der gegenständlichen Arbeit in Kapitel 11.2 durch und beantwortet die Forschungsfragen in Kapitel 11.3. Abgerundet wird dieses Kapitel in Abschnitt 11.4 mit einem Fazit sowie einem Ausblick der Autorin über weitere mögliche Forschungsvorhaben im gegenständlichen Themengebiet (siehe Kapitel 11.5).

11.1 Erzieltes Resultat

Die in Kapitel 1.4 als zu erwartende Ergebnisse vorgestellten Produkte der gegenständlichen Forschungsarbeit konnten von der Autorin erstellt werden. Neben den empirischen Erkenntnissen, die in den jeweiligen Kapiteln (Kapitel 8.6.2.6 und 8.6.2.7 für das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“, Kapitel 9.1.6 und 9.1.7 für die Evaluierung der Hardware, Kapitel 9.2.6 und 9.2.7 für die Evaluierung der Software sowie Kapitel 10.4.2.5, 10.4.4.5 und 10.5 für den Child-Centered-Design-Prozess) dargelegt sind und im Kapitel 11.2 bei der Hypothesenprüfung und bei der Beantwortung der Forschungsfragen in Kapitel 11.3 zusammengefasst betrachtet werden, sind als Innovationen das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ und ein adaptierter Child-Centered-Design-Ansatz, beide als Maßnahmen zur frühen Technikbildung in Informatik, entstanden.

11.1.1 Framework Informatik

Die Entwicklung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ beschreibt die Autorin in Kapitel 8. Hierzu arbeitete sie entlang eines vordefinierten Prozesses (siehe Kapitel 8.1) Leitthemen aus, die zum einen mit Lernzielen und zum anderen mit Lernergebnissen versehen sind (siehe Kapitel 8.2). Durch ein induktives Vorgehen erzeugte die Verfasserin der gegenständlichen Arbeit basierend auf konkreten Aufgaben (siehe Kapitel 8.4) verallgemeinerte Aktivitätsschablonen, die auf ihre Lernergebnisse hin überprüft (siehe Kapitel 8.6.2), in das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ (siehe Kapitel 8.7) aufgenommen wurden.

Aus ursprünglich 118 konkreten, evaluierten und empirisch belegten Aufgaben, die in fünf Inhaltsdimensionen zu neun Aufgabenarten und vier Schwierigkeitsgraden eingeordnet sind, entstand ein Framework zu drei Leitthemen („Early Childhood Technology“, „Computational Thinking“, „Child-Centered-Design“), mit 50 verallgemeinerten generalisierten Aktivitätsschablonen, die den neun Tätigkeiten einer Tätigkeitslandkarte (siehe Kapitel 8.5) zugeordnet sind. Auch die Aktivitätsschablonen des Frameworks lassen sich vier unterschiedlichen Schwierigkeitsgraden zuordnen. Das entstandene Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ zur frühen Technikbildung in Informatik bietet die Möglichkeit, aus den 50 verallgemeinerten Aktivitätsschablonen beliebige individuelle konkrete Aufgaben abzuleiten, die für den jeweiligen Einsatzkontext (Hard- und Software sowie Kindergarten) geeignet sind.

11.1.2 Adaptiertes Child-Centered-Design

Für den Einsatz von Child-Centered-Design (siehe Kapitel 10) erweiterte die Verfasserin der gegenständlichen Arbeit das in der Literatur bekannte Child-Centered-Design (es entspricht einem User-Centered-Design-Ansatz) um Cooperative-Inquiry und Bonded-Design.

In Zusammenarbeit mit den am Child-Centered-Design-Prozess teilnehmenden Kindern entwickelte die Autorin einen Prototyp einer Benutzerschnittstelle für eine Mikrowelt zum Programmieren lernen, die für 3- bis 6-jährige Kinder geeignet ist. Der finale Prototyp findet sich in Kapitel 10.4.3. Es handelt sich hierbei um eine Iteration des ersten Prototyps, den die Autorin im Rahmen einer Usability Evaluierung von Kindern der Zielgruppe testen ließ. Die Erkenntnisse aus diesem Test flossen wiederum direkt in die, eben genannte, Evolutionsstufe ein, die die Kinder ebenfalls testeten. Somit konnten die teilnehmenden

Kindergartenkinder im Rahmen der Child-Centered-Design-Aktivitäten die vier Rollen Benutzerinnen, Benutzer, Testerinnen, Tester, Informantinnen, Informanten sowie Gestaltungspartnerinnen und -partner (nach Nettet & Large, 2004) selbst einnehmen und Informatik, im Speziellen Software Entwicklung aus unterschiedlichen Sichten kennenlernen. Darüber hinaus konnte die Autorin durch die empirische Analyse des Child-Centered-Design-Einsatzes im Kindergarten aufzeigen, dass Child-Centered-Design in der Lage ist, einen Beitrag zur frühen Technikbildung in Informatik zu leisten (siehe Kapitel 10.5). Auch die Teilnahme an reinen Usability-Evaluierungen ohne Mitwirkung am Design-Prozess leistet einen Beitrag zur frühen Technikbildung in Informatik (siehe Kapitel 9.1.7.3 und 9.2.7.3).

11.1.3 Verbesserungspotentiale für Hard- und Softwareprodukte

Zusätzlich zur Prüfung der Hypothesen und Beantwortung der Forschungsfragen lassen sich aus den gewonnenen empirischen Erkenntnissen (siehe Kapitel 8.6.2.6, 10.4.2.5 und 10.4.4.5) sowie den Ergebnissen der Hard- und Software Evaluierung (siehe Kapitel 9.1.6 und 9.2.6) und als Ergänzung zur Literatur, siehe Liebal & Exner (2011), Pohl & Schmalzl (2010) Usability-Guidelines ableiten, an denen sich Hard- und Softwareprodukte, die zum Erlernen der Programmierung für 3- bis 6-jährige Kinder zum Einsatz kommen, orientieren sollten, um als zielgruppengerecht zu gelten:

- **Robustheit** - Hardware muss gegen Beschädigungen aller Art möglichst resistent sein, vor allem gegen das Einwirken von Kräften. Kindern kann schon einmal ein Gerät aus der Hand fallen, ohne dass dies mit Absicht geschieht. Software hingegen muss so entwickelt sein, dass auch eine vermehrte Fehlbedienung nicht zum Absturz führt.
- **Trial & Error** - Das grundlegende Prinzip, das Kinder beim Erkunden von Neuem einsetzen, muss von der eingesetzten Hard- und Software gut unterstützt werden. Fehlbedienungen dürfen nicht zum Absturz führen und eingegebene Befehle rückgängig zu machen muss genauso möglich sein.
- **Funktionsumfang** - Die eingesetzten Materialien sollten einen klar abgegrenzten und für Kinder gut überschaubaren Funktionsumfang aufweisen. Je eindeutiger für die Kinder erkennbar ist, wofür bzw. wie Materialien genutzt werden können, umso eher werden sie zielgerichtet von ihnen eingesetzt. Mit steigendem Alter sollte auch der Funktionsumfang der Materialien zunehmen, was im Rahmen von Software z.B. gut mit adaptiven Systemen, die sich am Alter der Kinder orientieren, abgebildet werden könnte. Je jünger die Kinder sind, desto einfacher und geringer ist der Funktionsumfang der Benutzerschnittstelle zu gestalten.
- **Rückmeldungen** - Sowohl Hard- als auch Software soll für Kinder stets eindeutige Rückmeldungen über die erfolgten Befehlseingaben geben. Nur so können Kinder wahrnehmen, dass ihre Befehle entgegengenommen wurden. Zusätzlich müssen die Rückmeldungen zeitnah und entsprechend ihrer Wertung (positiv, neutral, negativ) erfolgen. Positive Rückmeldungen sollen z.B. beim korrekten Lösen einer Aufgabe erscheinen, neutrale z.B. bei der Eingabe von Befehlen und negative z.B. beim falschen Lösen von Aufgaben oder bei Fehlbedienungen.
- **Gestaltung** - Die einzusetzenden Materialien sollten kindgerecht gestaltet sein. Dazu sind idealerweise freundliche, eher warme Farben zu wählen. Lächelnde Gesichter auf Figuren wirken motivierend und begeisternd auf Kinder und Tiere als Spielfiguren sind bewährt.

Software sollte darüber hinaus über folgende zusätzliche Eigenschaften verfügen:

- **Navigation** - Für die Kinder, die die Software zum Erlernen der Programmierung nutzen muss stets klar ersichtlich sein, an welchem Punkt sie sich im Lernprogramm befinden. Insbesondere bei der Bearbeitung von Aufgaben muss stets klar sein, was als Nächstes zu tun ist, welche Aufgaben folgen und wie die vorhergegangene

Aufgabe aufgerufen werden kann. Auch eine problemlose Rückkehr zum Ausgangspunkt (Startbildschirm) sollte stets gegeben sein.

- Steuerung / Bedienung - Die Benutzerschnittstelle sollte über wenige, dafür große Buttons zur Bedienung der Software verfügen. Die Buttons sollten Icons beinhalten, die selbsterklärend sind und die Bedienung der Benutzerschnittstelle bzw. der Software muss bei der Zielgruppe der 3- bis 6-jährigen Kinder ohne den Einsatz von Text auskommen, das gilt insbesondere für Aufgaben- und Hilfestellungen sowie Fehlermeldungen.
- Animationen - Lernsoftware sollte wenige Animationen beinhalten. Für Kinder im Alter von 3 bis 6 Jahren sind Animationen irritierend, sie verstehen oftmals nicht, warum das Programm gerade eben etwas ausführt, obwohl sie nichts eingegeben haben. Je älter Kinder sind, desto eher können sie Animationen als solche wahrnehmen.
- Positionierung - Kinder nehmen verstärkt zuerst die Mitte des Bildschirms wahr. Zentrale Elemente der Benutzerschnittstelle sind daher mittig zu platzieren. Elemente, die am Rand des Bildschirms positioniert sind, werden von 3- bis 6-jährigen Kindern oft nur schwer erkannt oder es bedarf expliziter Hilfestellungen, damit die Kinder diese Bereiche bewusst wahrnehmen.

11.2 Überprüfung der Hypothesen

11.2.1 H1.1

H1.1: Wenn 3- bis 6-jährige Kinder altersgerecht in der Programmierung einfacher, grundlegender Befehlsabfolgen geschult werden, dann können sie Problemstellungen desselben Schwierigkeitsgrades nach Teilnahme an Lehr-Lern-Einheiten, die aus dem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ abgeleitet werden können, selbständig lösen.

Diese Hypothese kann nur unter der Bedingung angenommen werden, dass den 3- bis 6-jährigen Kindern zuerst die Hardware (hier der Bee-Bot) und anschließend die Software (hier Focus on Bee-Bot) näher gebracht wird. Dies zeigt sich im Abschluss-Assessment, siehe Kapitel 8.6.2, welches von der Autorin durchgeführt wurde, um die gegenständliche Hypothese zu prüfen. Dazu setzte die Autorin zwei Gruppen ein. Die Gruppe ASG1 lernte zuerst den Hardware Bodenroboter und seine Programmierung kennen, sowie anschließend die Programmierung der Softwaresimulation des Bodenroboters. Im Gegensatz dazu lernte die Gruppe ASG2 zuerst die Programmierung der Softwaresimulation des Bodenroboters gefolgt von der Programmierung des Hardware Bodenroboters kennen. In beiden Gruppen wurde die gleiche Hard- und Software eingesetzt.

Die Ergebnisse des Abschlusstests unterscheiden sich bei diesen beiden Gruppen. Während die Gruppe ASG1 beim Abschluss-Assessment gut abschnitt, zeigte die Kontrollgruppe ASG2, bei der die Autorin überprüfte, ob die Reihenfolge der Aktivitäten eine Auswirkung auf die Ergebnisse des Abschluss-Assessments hat, deutliche Schwächen.

Bei einfachen Aufgaben (siehe Kapitel 8.6.2.6.5) konnten fast alle teilnehmenden Kinder aus der Gruppe ASG1 auf Anhieb eine adäquate Lösung präsentieren. In der Gruppe ASG2 gelang dies nur rund der Hälfte der Kinder. In der Gruppe ASG1 bedienten sich deutlich weniger Kinder eigener Hilfestellungen, wie z.B. dem vorab Auszählen der benötigten Schritte oder „Trockenweiterstellen“.

Mit zunehmender Komplexität und Schwierigkeit der gestellten Aufgaben sank die Leistung der teilnehmenden Kinder in der selbständigen Lösung derselben. Gelegentlich musste die Autorin hier Hilfestellungen geben, damit die Kinder das gewünschte Ziel mit dem Bodenroboter erreichten. Den teilnehmenden Kindern aus der Gruppe ASG1 gelang es großen Teils komplexe Wege zu programmieren und rund die Hälfte nutzte dabei einen kürzesten Weg. Manche bedienten sich hierbei der Technik „Trockenweiterstellen“ (siehe Kapitel 8.3.2). In der Gruppe ASG2 nutzten alle Kinder die Möglichkeiten des

„Trockenweiterstellens“. Während rund ein Drittel der Kinder der Gruppe ASG2 laut erklärte, welche Befehle sie programmieren, tat dies in der Gruppe ASG1 rund die Hälfte. In der Gruppe ASG2 musste der Versuch, dem Bee-Bot einen komplexen Weg einzuprogrammieren bei mehr als der Hälfte der Kinder abgebrochen werden.

An der Aufgabenstellung, einen komplexen Weg ohne Einzelschritte bzw. ohne „Trockenweiterstellen“ zu programmieren, scheiterte die Gruppe ASG2, wobei nur ein teilnehmendes Kind versuchte, diese Aufgabe zu lösen. Wie in Kapitel 8.6.2.6.8 dargelegt, lösten rund 85 % der Kinder der Gruppe ASG1 diese Aufgabe. Auch die grundsätzliche Bedienung des Hardware Bodenroboters war für die Kinder der Gruppe ASG2 eine größere Herausforderung als für die Gruppe ASG1. Weitere Unterschiede sind der Beantwortung der Hypothese H1.4 (siehe Kapitel 11.2.4) zu entnehmen.

Diese gesammelten Erkenntnisse hat die Autorin in die Gestaltung des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“, siehe Kapitel 8.7, einfließen lassen.

Somit ist festzuhalten: Die Hypothese H1.1 ist anzunehmen, wenn Kinder zuerst den Hardware Bodenroboter kennenlernen und anschließend die Softwaresimulation des Bodenroboters. Die Hypothese ist jedoch zu verwerfen, wenn die geforderte Reihenfolge, zuerst den Hardware Bodenroboter kennenzulernen und erst im Anschluss die Softwaresimulation einzusetzen, nicht gewährleistet wird.

11.2.2 H1.2

H1.2: Wenn Kinder im Alter von 3 bis 6 Jahren in der Programmierung eines Bodenroboters geschult werden, dann können sie selbständig ohne zusätzliche Einführung die eingesetzte Programmierumgebung / Mikrowelt des virtuellen Bodenroboters am PC (=Simulation) bedienen, um einfache, grundlegende Befehlsabfolgen auszuführen.

Zur Prüfung dieser Hypothese setzte die Autorin eine Evaluierung der Usability der Softwaresimulation der Mikrowelt (hier: Software Focus on Bee-Bot) ein, siehe Kapitel 9.2. Mit den Ergebnissen dieser Evaluierung kann die Prüfung der gegenständlichen Hypothese abgeleitet werden, zusätzlich können weitere Hypothesen geprüft und Forschungsfragen beantwortet werden. Hierzu ist die Gruppe SWT1 zu untersuchen, in der die teilnehmenden Kinder zuerst den Hardware Bodenroboter (hier: Bee-Bot) kennenlernten und anschließend die Softwaresimulation nutzten.

Die Hypothese H1.2 ist zu verwerfen, da, wie in Kapitel 9.2.6.3 dargelegt, kein Kind in der Softwaresimulation ohne Hilfestellung erkannte, dass der virtuelle Bodenroboter durch Betätigung von (Richtungs-)Buttons, die in einem eigenen Navigationsbereich angesiedelt sind, programmiert wird. Diese Art der Programmierung steht im Gegensatz zum realen Hardware Bodenroboter, der durch Betätigung von Tasten am Rücken programmiert wird. Das Programm suggeriert den Kindern durch seine Gestaltung der zentralen Figur mit ihren aufgemalten Tasten am Rücken eine weitere Bedienungsmöglichkeit als die ursprünglich vorgesehene Eingabe von Richtungsbefehlen via Mausklicks auf abgebildete Richtungs-Buttons im Eingabebereich links oben. Dies führte zu Beginn bei den Kindern zu Irritationen. (Siehe Kapitel 9.2.7) Rund zwei Drittel der teilnehmenden Kinder waren zu Beginn der Meinung, dass das Programm mit anderen Eingabegeräten als der Maus, wie z.B. Tastatur, TouchScreen, o.ä. zu bedienen ist. (Siehe Kapitel 9.2.7.1)

Auch das Auffinden des GO-Buttons zum Losfahren des virtuellen Bodenroboters in der Softwaresimulation stellte die Kinder vor eine Herausforderung. (Siehe Kapitel 9.2.6.3.1) Nach dem Erlernen der Grundlagen der Bedienung der Softwaresimulation und Programmierung des virtuellen Bodenroboters, hatten die Kinder erste Erfolge zu verzeichnen.

Dennoch, knapp der Hälfte der teilnehmenden Kinder gelang es, ihr Wissen aus der Bedienung und Programmierung des Hardware Bodenroboters auf die Bedienung bzw.

Programmierung des virtuellen Bodenroboters in der Mikrowelt zu übertragen. (Siehe Kapitel 9.2.6.9.1)

Somit ist festzuhalten: Die Hypothese H1.2 ist zu verwerfen. Ohne weitere Einführung ist es 3- bis 6-jährigen Kindern, die in der Programmierung eines Bodenroboters (hier: Bee-Bot) geschult wurden, nicht möglich, selbständig die eingesetzte Programmierumgebung / Mikrowelt des virtuellen Bodenroboters am PC (hier: Software Focus on Bee-Bot) zu bedienen und so einfache, grundlegende Befehlsabfolgen auszuführen.

11.2.3 H1.3

H1.3: Wenn 3- bis 6-jährige Kinder in der Bedienung der zum Programmieren eingesetzten zielgruppengerechten Hard- und Softwareprodukte geschult werden, dann können sie Usability-Schwächen derselben, die mit einem Expertinnen- und Experten-Review korrespondieren, identifizieren.

Diese Hypothese ist zu verwerfen. 3- bis 6-jährige Kinder, die in der Bedienung der zum Programmieren eingesetzten zielgruppengerechten Hard- und Softwareprodukte geschult werden, können keine Usability-Schwächen derselben, die mit einem Expertinnen- und Experten-Review korrespondieren, identifizieren. 3- bis 6- Jährige können grundsätzlich nur wenige bis gar keine Usability-Schwächen identifizieren, da sie zu großen Teilen Schwierigkeiten damit haben, ihre gewonnenen Erkenntnisse zu verbalisieren und zu vermitteln, siehe Kapitel 9.1.7. Kinder sind zwar experimentierfreudig und an Neuem sehr interessiert, aber nur rund die Hälfte der Kinder, die z.B. an der Evaluierung der Hardware Usability teilnahmen, interagierte ansatzweise unterstützend bei der Auffindung von Usability-Schwächen mit der Testleitung. Die Thinking Aloud Komponente des Cognitive Walkthrough stellte sich als eine Herausforderung im Testsetting dar, was auch in der Literatur berichtet wird, siehe Kapitel 5.3.1. Ständige Motivation der Kinder durch die Testleitung zur Beantwortung von Fragen oder zum Erzählen, was die Kinder im Moment machen, erschwerte eine objektive Analyse.

Die Evaluierung der Software Usability, siehe Kapitel 9.2.7, legt dar, dass Kinder zwar Usability-Schwächen aufzeigen bzw. auf Usability-Schwächen stoßen können, diese aber nicht bewusst als solche wahrnehmen. Die Beurteilung, ob es sich um eine Usability-Schwäche oder eine Fehlbedienung seitens des Kindes handelt, obliegt einzig und allein den begleitenden und testdurchführenden Expertinnen und Experten.

Somit ist festzuhalten: Die Hypothese H1.3 ist zu verwerfen.

11.2.4 H1.4

H1.4: Wenn Kinder im Alter von 3 bis 6 Jahren, unter der Voraussetzung, dass in beiden Fällen das gleiche Lehr-Lern-Einheiten-Programm zum Einsatz kommt, bei dem nur die Reihenfolge der Aktivitäten verändert wird, Programmieren lernen, in dem sie zuerst einen Hardware Bodenroboter und anschließend eine korrespondierende Softwaresimulation (Programmierumgebung / Mikrowelt) nutzen, dann erzielen diese Kinder bessere Ergebnisse bei einem Assessment als die Kinder, die zum Programmieren lernen zuerst die Softwaresimulation (Programmierumgebung / Mikrowelt) und anschließend den korrespondierenden Hardware Bodenroboter nutzen.

Wie die Kapitel 9.1.7.1 und 9.2.7.1 sowie die Prüfung der Hypothesen H1.1 und H1.2 zeigen, besteht ein Unterschied in der Leistung der Kinder in Abhängigkeit, ob diese zuerst mit dem Hardware Bodenroboter (Gruppe HWT1, SWT1 bzw. ASG1) oder mit der Softwaresimulation (Gruppe SWT2, HWT2 bzw. ASG2) programmieren gelernt haben.

Zu berücksichtigten ist hier jedoch das Ergebnis der Hypothese H1.2 der gegenständlichen Arbeit, dass Kinder Unterstützung durch Tutorinnen und Tutoren beim Wissenstransfer zwischen Hardware und Software benötigen.

Mit Hilfe der Studie, bestehend aus der Evaluierung der Hardware Usability (Kapitel 9.1), der Evaluierung der Software Usability (Kapitel 9.2) und dem Abschluss-Assessment beim Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ (Kapitel 8.6.2), zeigt die Autorin auf, dass 3- bis 6-jährige Kinder in der Lage sind, selbständig einen Bodenroboter zu programmieren und mit ihm ein vorher definiertes Ziel zu erreichen. Wie die Ergebnisse des Abschluss-Assessments nach Besuch der Lehr-Lern-Einheiten der Studie zeigen (siehe Kapitel 8.6.2.6.5 bis 8.6.2.6.7) sind 3- bis 6-jährige Kinder der Gruppe ASG1 mit hoher Wahrscheinlichkeit in der Lage nicht nur einfache Problemstellungen sehr gut, sondern sogar komplexere mit dem Bodenroboter zu lösen. Die Kinder der Gruppe ASG2 haben hierbei Schwierigkeiten. Eine durchgehende Programmierung, ohne Einzelschritte („Trockenweiterstellen“, siehe Kapitel 8.3.2), stellt die Kinder der Gruppe ASG2 gegenüber den Kindern der Gruppe ASG1 jedoch vor erhebliche Probleme (siehe Kapitel 8.6.2.6.8). Exemplarische Unterschiede der beiden Gruppen sind in Kapitel 8.6.2.7.2 zu finden.

Aus der Beobachtung der beiden Gruppen ASG1 und ASG2 lässt sich ableiten, dass die Gruppe ASG1, die zuerst die Hardware und anschließend die Software kennenlernte, in Summe die besseren Ergebnisse aufweist. Die Gruppe ASG2 hingegen, die zuerst die Software und dann die Hardware kennenlernte, hatte im Rahmen des Abschluss-Assessments deutliche Probleme bei der Programmierung der Hardware.

Beide Gruppen, ASG1 und ASG2, lernten unter Einsatz des gleichen Lehr-Lern-Einheiten-Programms programmieren, wobei nur die Reihenfolge der Aktivitäten verändert wurde und die aufgewendete Zeit für die jeweiligen Aktivitäten in beiden Gruppen gleich war.

Auch wenn diese ersten empirischen Befunde keinesfalls repräsentativ und mit Vorsicht zu behandeln sind, so lässt sich erkennen, dass sich Kinder mit der Programmierung in einer Programmierumgebung am PC (= eine Mikrowelt) leichter tun, wenn sie zuerst programmieren mit Hilfe von Hardware Bodenrobotern lernen. Kinder hingegen, die programmieren mit Hilfe einer Mikrowelt lernen und anschließend ähnliche Programmierungen mit Hilfe von einem Hardware Bodenroboter vornehmen sollen, fällt der Wissenstransfer von der Software zur Hardware schwerer.

Somit ist festzuhalten: Die Hypothese H1.4 ist anzunehmen.

11.2.5 H2.1

H2.1: Wenn Child-Centered-Design mit 3- bis 6-jährigen Kindern als Methode zur Entwicklung eines Prototyps einer grafischen Benutzerschnittstelle für eine zielgruppengerechte Mikrowelt eingesetzt wird und Kinder in den gesamten Design-Prozess involviert werden, dann entsteht ein Prototyp, der als Basis für die Entwicklung einer zielgruppengerechten Mikrowelt dienen kann und dessen Bedienelemente von 3- bis 6-jährigen Kindern, die nicht am Design-Prozess teilgenommen haben, selbständig und ohne Einführung genutzt werden können.

Die durchgeführten Evaluierungen des Prototyps 1, siehe Kapitel 10.4.2 und des Prototyps 2, siehe Kapitel 10.4.4, werden von der Autorin zur Prüfung der Hypothese herangezogen. Im adaptierten Child-Centered-Design-Prozess waren 5- bis 6-jährige Kinder aktiv, überwiegend in der Rolle der Informantinnen und Informanten, siehe Kapitel 5.2.1. Es zeigt sich, dass 5- bis 6-jährige Kinder den Prototyp sinngemäß und weitestgehend selbstständig nutzen können, siehe Kapitel 10.4.4.5.

3- bis 4-jährige Kinder wurden auf Grund der zuvor gewonnenen Erfahrungen bei der Durchführung der Lehr-Lern-Einheiten (siehe 8.3) von der Autorin nicht zur gesamten Teilnahme am adaptierten Child-Centered-Design-Prozess eingeladen, da die Kinder dieser Altersklasse oftmals beim Zeichnen daneben sitzen und warten, bis andere etwas gezeichnet haben, was sie dann nachzeichnen können oder sie zeichnen Inhalte, die nicht mit der Aufgabenstellung in Zusammenhang stehen. Auch bei dem Software-Usability-Test (siehe Kapitel 9.2.6) und dem Hardware-Usability-Test (siehe Kapitel 9.1.6) sowie beim Abschluss-Assessment (siehe Kapitel 8.6.2.6) hat sich gezeigt, dass sich die ganz jungen

Kinder, auf Grund ihrer noch fehlenden Vorerfahrung, oftmals noch etwas schwer tun Inhalte verbal wie auch non-verbal zu kommunizieren.

Die Evaluierung des Prototyps mit 3- bis 4-jährigen Kindern als Testerinnen und Tester zeigte, siehe Kapitel 10.4.4.5 und 10.4.4.6, dass diese den Prototyp nur mit Unterstützung durch Tutorinnen oder Tutoren sinngemäß nutzen können. Ihnen gefällt jedoch die Benutzeroberfläche und sie äußerten den Wunsch, damit spielen zu wollen.

Schlussendlich unterstützen diese Erkenntnisse die Vermutung, dass ein adaptierter Child-Centered-Design-Prozess mit aktiver Beteiligung durch Kinder der Zielgruppe die Qualität und Benutzerfreundlichkeit einer grafischen Benutzeroberfläche für eine zielgruppengerechte Software steigert und die Bedienung selbiger erleichtert.

Somit ist festzuhalten: Die Hypothese H2.1 ist zu verwerfen, da 3 bis 4-jährige Kinder nicht aktiv an einem Child-Centered-Design-Prozess teilnehmen können. Mit der Einschränkung der Altersgruppe auf 5- bis 6-jährige Kinder ist die Hypothese anzunehmen.

11.3 Beantwortung der Forschungsfragen

Es stehen am Markt seit fast 40 Jahren Programmierumgebungen für junge Kinder zur Verfügung. Jedoch sind nur wenige für die Zielgruppe der 3- bis 6-jährigen Kinder geeignet. In Kapitel 4.3 werden Lernumgebungen zum Programmieren für Kinder vorgestellt. Hierbei ist zwischen Programmierumgebungen und Mikrowelten zu unterscheiden (siehe Kapitel 4.3.1.1). Mikrowelten eignen sich für einen Einsatz mit Kindern besonders, da sie darauf ausgelegt sind, den Unterschied zwischen den mentalen Modellen der Lernenden und der Programmiersprache zu verringern. Oftmals treten Mikrowelten in Kombination mit Mini-Languages (siehe Kapitel 4.3.1.2) auf, die zur Unterstützung beim Erlernen von Programmiersprachen auf ein einfaches und kleines Befehlsset zurückgreifen. Mini-Languages fördern das systematische Lösen von Problemen und das algorithmische Denken. Eine der bekanntesten Mini-Languages und zugleich auch Programmierumgebungen ist LOGO, die von Seymour Papert am MIT entwickelt wurde (siehe Kapitel 2.1.4). Jedoch ist LOGO in ihrem vollen Umfang für die Zielgruppe der 3- bis 6-jährigen Kinder eher ungeeignet. Für die Zielgruppe geeignet sind ToonTalk, Kidsim & Cocoa, Stagecast Creator, ScratchJr und Focus on Bee-Bot (siehe Kapitel 4.3.1.3) Die Software Focus on Bee-Bot ist die graphische Simulation des Bodenroboters Bee-Bot (siehe Kapitel 4.4.2), der über eine Mini-Language, basierend auf Grundelementen von LOGO, zur Programmierung verfügt.

Darüber hinaus stehen sog. Tangible Interfaces (siehe Kapitel 4.3.2) zur Verfügung, insbesondere Tangicons, Electronic Blocks, Kiwi, Kibo, Tern und Cherp erscheinen für einen Einsatz mit Kindergartenkindern geeignet. Aus der Forschungsrichtung „Educational Robotics“ (siehe Kapitel 4.4) abgeleitet, werden für Kinder im Alter von 3 bis 6 Jahren der Roamer, Curlybot, spezielle Lego® Roboter oder der Bee-Bot, Constructa-Bot, Pro-Bot oder auch Blue-Bot empfohlen.

Die Rahmenbedingungen in österreichischen Kindergärten sind zur Durchführung von Aktivitäten zur frühen Technikbildung im Bereich der Informatik differenziert zu betrachten. Positiv hervorzuheben ist, dass sowohl das Betreuungspersonal in den Kindergärten als auch die Kinder und deren Eltern sehr interessiert an den entsprechenden Aktivitäten sind, solange dafür kein zusätzliches finanzielles Budget von Nöten ist. Dem gegenüber stehen Aspekte, die es mittelfristig zu beheben gilt: Es bedarf mehr (zur frühen Technikbildung) ausgebildeten Personals, technischen Equipments und finanzieller Mittel.

Als eine Herausforderung, in Bezug auf die Durchführung der Studie, offenbarte sich die schwankende Anwesenheit der Kinder. Da in der Vorstudie (im Vorfeld der gegenständlichen Forschungsarbeit, siehe Kapitel 1.4) die Durchführung der Lehr-Lern-Einheiten über mehrere Wochen (exkl. Feiertagen und Ferienzeiten) mit jeweils einer Einheit pro Woche geplant wurde, waren z.B. auf Grund von Krankheit nicht immer alle Kinder anwesend. Auch wenn bei der gegenständlichen Studie aufgrund dieser Erfahrungen die Lehr-Lern-Einheiten

geblockt an fünf aufeinanderfolgenden Tagen stattfanden, um der Schwankung der Anwesenheit entgegenzuwirken, waren nicht immer alle Kinder anwesend – im direkten Vergleich war die Anwesenheit höher, woraus zu schließen ist, dass eine geblockte Abhaltung zu empfehlen ist. Es ist nach der Durchführung der Lehr-Lern-Einheiten unter Einsatz des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ die Tendenz erkennbar, dass es möglich ist, dass Kinder in weniger als zehn Lehr-Lern-Einheiten in der Lage sind, die Programmierung eines Bodenroboters zu erlernen.

Die Betreuung (z.B. Toilettengang, Schneuzen, etc) der Kinder während der Lehr-Lern-Einheiten stellt eine zusätzliche Herausforderung dar, da Kindergärten für zusätzliche Lehr-Lern-Einheiten kaum Betreuungspersonal zur Verfügung stellen können, lediglich die Aufsichtspflicht bleibt gewahrt. Wenn Betreuungspersonal zur Verfügung stand, waren die Kinder immer wieder irritiert, wer für den Zeitraum der Lehr-Lern-Einheiten „das Sagen hat“ - das Betreuungspersonal des Kindergartens oder die Programmleitung, was zur Ablenkung der Kinder vom Inhalt der Aktivitäten führte. Die Autorin berücksichtigte diesen Aspekt für ihre Studie und setzte eigenes Betreuungspersonal für die Kinder ein, so dass sich die Kinder auf die Aufgaben in den Lehr-Lern-Einheiten konzentrieren konnten und das Kindergartenpersonal sich darauf verlassen konnte, dass die Kinder entsprechend betreut werden.

Grundsätzlich sind Programme zur frühen Technikbildung im Bereich der Informatik in Kindergärten in Österreich durchführbar. Besonders zu berücksichtigen ist, dass ausreichend Platz für die Aktivitäten sowie Betreuungspersonal zur Unterstützung der Programmleitung, damit diese sich auf die Inhalte konzentrieren kann, zur Verfügung stehen. Die gegenständliche Studie hat gezeigt, dass ein Betreuungsverhältnis von ca. 1:3 bis 1:4, also auf eine Betreuungsperson kommen drei, maximal vier Kinder, als ideal anzusehen ist. In diesem Fall wird die Programmleitung zu den Betreuungspersonen hinzugezählt.

Die Programmleitung selbst muss für eine frühe Technikbildung in Informatik über ausreichende fachdidaktische sowie fachthematische Kompetenzen verfügen und das Betreuungspersonal sollte zumindest grundlegendes Fachwissen bezüglich der behandelten Inhalte aufweisen, um die Programmleitung ideal unterstützen zu können. Dies erfordert vom Kindergartenpersonal die Bereitschaft zur entsprechenden Weiterbildung. (Siehe dazu Kapitel 2.3, im Speziellen Kapitel 2.3.2) Eine andere Möglichkeit zur Wissensvermittlung im Bereich der Informatik im Kindergarten ist der Einsatz von (zum Kindergarten) externen Kräften, die die Inhalte vermitteln. Idealerweise sind diese Kräfte mit den notwendigen Materialien ausgerüstet und bringen diese an den Veranstaltungsort mit. Das muss jedoch finanziell abgegolten werden – entweder durch Sponsoren, Kursgebühren oder durch die Unterstützung der Betreiberinnen und Betreiber der Kindergärten.

Für die teilnehmenden Kinder sollte ausreichend Material zur Verfügung stehen. Aus den gewonnenen Erfahrungen ist abzuleiten, dass für maximal zwei Kinder in einem Team jeweils ein Bodenroboter zur Verfügung stehen muss, sonst schwindet das Interesse bei den Gruppenaufgaben auf Grund zu langer Wartezeiten der anderen Gruppenmitglieder. Weiter ist darauf hinzuweisen, dass die Aktivitäten nicht zu lange dauern dürfen, abwechslungsreich gestaltet und bezüglich Sozialform, Art der Aufgabe und Methode variiert werden müssen, damit die Aufmerksamkeit der teilnehmenden Kinder erhalten bleibt.

Für eine Durchführung des Programms zur frühen Technikbildung in Informatik halten sich die notwendigen finanziellen Investitionen im Rahmen; es werden lediglich eine ausreichende Anzahl an Bodenrobotern (hier: Bee-Bots) benötigt, sowie einige Spielmatte nach Wunsch und Bastelmaterialien. Eventuell ergänzen noch übergroße Spielkarten mit aufgezeichneten Befehlen (Befehlskarten) die Materialien, um den Kindern eingegebene Befehlsabfolgen visualisieren zu können. Diese können mit etwas Geschick auch selbst gebastelt und müssen nicht extra käuflich erworben werden. Typisches Bastelmaterial (Scheren, Stifte, Papier, etc.) ist in den Kindergärten in der Regel vorhanden. Als Einstieg erscheinen diese Investitionen ausreichend und in Kombination mit dem entwickelten

Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“, siehe Kapitel 8.7, bieten sich, bei Einsatz von Kreativität zur Ableitung von Aufgaben aus den Aktivitätsschablonen, unzählige Möglichkeiten zur Weiterführung, ohne allzu große finanzielle Aufwände betreiben zu müssen. Bei der Durchführung der Aufgaben ist darauf zu achten, dass nach Möglichkeit die Kinder regelmäßig anwesend sind, um den Lernerfolg optimal zu unterstützen.

Wenn eine geeignete Software zum Einsatz kommen soll, dann ist entsprechende IT Hardware anzuschaffen. Hier ist vor allem auf eine robuste Gestaltung selbiger Wert zu legen, da Kinder manchmal, ungewollt, etwas harscher mit den Geräten umgehen. Gerade dann, wenn etwas nicht gleich so funktioniert, wie sie es vermutet haben.

11.3.1 F1.1

F1.1: Wie können mögliche didaktische Prinzipien und Aktivitätsschablonen aussehen, im Zuge derer Kinder die Programmierung einfacher grundlegender Befehlsfolgen erlernen können?

Wie in Kapitel 8 dargestellt entwickelte die Verfasserin der gegenständlichen Arbeit ein Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“, welches Inquiry Based Learning und Discovery Learning (beide siehe Kapitel 2.3.1) als Integrationsmethode von IKT in Lehr-Lern-Szenarien nutzt. Die Entwicklung des Frameworks basiert u.a. auf Anregungen und Inspirationen des CSTA-Frameworks Level 1 (siehe Kapitel 3.1.2), der „four P’s of Creative Learning“ von Resnick (2014) (siehe Kapitel 3.1.2), des sechs C-Modells (siehe Kapitel 3.1.2), der Grundlagen informatischer Bildung nach Hubwieser (2007) (siehe Kapitel 3.1), der Bildungswerte der Informatik nach Futschek (2003) (siehe Kapitel 3.1.2) und des PLU-Modells von Markopoulos et al. (2008) (siehe Kapitel 5.1.1.1).

Das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ beinhaltet drei Leitthemen („Early Childhood Technology“, „Computational Thinking“, „Child-Centered-Design“; siehe Kapitel 8.2), mit 50 verallgemeinerten generalisierten Aktivitätsschablonen, die neun Tätigkeiten einer Tätigkeitslandkarte und vier Schwierigkeitsgraden zugeordnet sind. Das Framework ist in Kapitel 8.7 zu finden.

Die Entwicklung der Aktivitätsschablonen basiert auf einer Abstraktion von konkreten Aufgaben, die aus Aufgabenentwürfen (siehe Kapitel 8.2.5) entstanden, die sich zum einen an den Darlegungen zur Kompetenzentwicklung durch Einsatz von IT und Robotern (siehe Kapitel 3.2), zum anderen an der kognitiven Entwicklung nach Piaget (siehe Kapitel 2.1.3) sowie an weiteren Erkenntnissen zur kognitiven und motorischen Entwicklung von Kindern der Zielgruppe (siehe Kapitel 5.1.1) orientieren. Besonderes Augenmerk schenkte die Autorin dem algorithmischen Denken (siehe Kapitel 3.1.4), dem Higher-Order-Thinking und dem Computational Thinking (siehe Kapitel 3.3).

Die verallgemeinerte Formulierung der Aktivitätsschablonen des Frameworks erlaubt den Einsatz eines beliebigen Bodenroboters und einer entsprechenden Softwaresimulation / Mikrowelt zu selbigem. Aus dem Framework können individuelle Aufgaben abgeleitet und mit 3- bis 6-jährigen Kindern durchgeführt werden. Als Input zur Ideenfindung konkreter Aufgaben, die basierend auf dem Framework entwickelt werden können, stehen in Anhang A2 die von der Autorin im Zuge der Lehr-Lern-Einheiten in zwei Kindergärten durchgeführten Aufgaben zur Verfügung.

11.3.2 F1.2

F1.2: Gelingt 3- bis 6-jährigen Kindern der Lerntransfer der mit dem Bodenroboter erlernten Programmierbefehlsabfolgen auf die Programmierumgebung / Mikrowelt des virtuellen Bodenroboters (=Simulation)?

Die Evaluierung der Software Usability und die Prüfung der Hypothese H1.2 zeigen, dass Kinder mit der Programmierumgebung / in der Mikrowelt des virtuellen Bodenroboters programmieren können, wenn sie vorher den Hardware Bodenroboter (hier: Bee-Bot, der über eine Mini-Language verfügt) kennen und programmieren gelernt haben, jedoch

benötigen die Kinder Unterstützung bei der Bedienung der Programmierumgebung / Mikrowelt.

Generell fiel den Kindern, die den Hardware Bodenroboter schon kannten, die Orientierung in und die Nutzung der Softwaresimulation des virtuellen Bodenroboters leichter als den Kindern, die diesen noch nicht kennengelernt hatten (siehe Kapitel 9.2.7). Kapitel 9.2.6.4 belegt, dass Kinder, die den Hardware Bodenroboter nicht kannten, erhebliche Probleme bei der sinngemäßen Nutzung des virtuellen Bodenroboters hatten. Dies ist unabhängig von sonstigen relevanten Vorkenntnissen zur Computernutzung zu betrachten, die in der Altersgruppe der 3- bis 6-Jährigen durchaus als gut bezeichnet werden können, siehe Kapitel 9.2.6.1.

Somit ist festzuhalten, dass der im Fokus des Interesses dieser Forschungsfrage F1.2 stehende Lerntransfer zwar gelingt, die Kinder allerdings durch die Unterschiede in der Bedienung des virtuellen zum realen Bodenroboter (siehe Kapitel 9.2.6.9) auf eine Einführung in die Bedienung des virtuellen Bodenroboters in der Software angewiesen sind.

11.3.3 F1.3

F1.3: Können Kinder im Alter von 3 bis 6 Jahren Usability-Schwächen von zielgruppengerechten Bodenrobotern und Programmierumgebungen / Mikrowelten am PC identifizieren?

Kinder können keine Usability-Schwächen als solche identifizieren, wie aus den Kapiteln 9.1 und 9.2 sowie der Prüfung der Hypothese 1.3 abgeleitet werden kann. Grund dafür ist, dass es den Kindern zumeist nicht möglich ist, eine Usability-Schwäche zu benennen, jedoch können sie durch ihre Interaktion mit dem jeweiligen Produkt einer Usability-Expertin bzw. einem Usability-Experten mögliche Usability-Schwächen aufzeigen.

Mögliche Verbesserungspotentiale für Benutzerschnittstellen, die für die Zielgruppe der 3- bis 6-jährigen Kinder geeignet sind, finden sich in Kapitel 11.1.3. Als zwei der wichtigsten Verbesserungspotentiale sind anzuführen, dass Benutzerschnittstellen für 3- bis 6-jährige Kinder so gut wie möglich auf Text verzichten sollten, da die Zielgruppe oft noch nicht lesen kann. Darüber hinaus muss die Programmierumgebung / Mikrowelt bei Aufgabenstellungen Hilfe anbieten, wie Aufgaben zu lösen sind, damit die Kinder, zumindest bis zu einem gewissen Grad, selbständig und autark ihre Fertigkeiten trainieren können und nicht durch ein Scheitern bei der Aufgabenlösung die Lust am Programmieren verlieren.

11.3.4 F1.4

F1.4: Sind geschlechtsspezifische Unterschiede hinsichtlich der Herangehensweise, des generellen Interesses und der Begeisterung beobachtbar?

Aus den durchgeführten Evaluierungen der eingesetzten Hardware und Software sowie dem Abschluss-Assessment sind nur vereinzelte geschlechtsspezifische Unterschiede auszumachen, die jedoch aufgrund der geringen Grundgesamtheit mit Vorsicht zu betrachten sind. Verallgemeinerungen können folglich nicht getroffen werden.

Wie in Kapitel 8.6.2.7.3 zusammengefasst, scheint der reale Bodenroboter, unabhängig vom Geschlecht, für Anfängerinnen und Anfänger in der Programmierung geeignet. Mädchen versuchen ihre Aufgaben wohl überlegt zu lösen und reden gerne über ihre Lösungsansätze. Hingegen versuchen die Jungen ihre Aufgaben eher mit dem Trial & Error Prinzip zu bearbeiten.

Bei der Evaluierung der Hardware Usability war zu beobachten, dass Mädchen bei diesem Testsetting in Bezug auf das Ausprobieren, die Herangehensweise und die Bedienung des Hardware Bodenroboters (hier: Bee-Bot) zurückhaltender agierten als Jungen. Bei der generellen Zurückhaltung (erzählen, was man im Moment macht) im Testsetting ist das Verhältnis Mädchen zu Jungen ausgeglichen, jedoch interagierten Mädchen eher mit der Testleitung als Jungen. (Siehe Kapitel 9.1.7.2)

Aus dem direkten Vergleich der beiden Geschlechter bei der Evaluierung der Software Usability, siehe Kapitel 9.2.7.2, lässt sich ableiten, dass Jungen eher experimentierfreudig sind als Mädchen, weniger der Problemstellung folgen und sich sehr leicht durch andere Elemente am Bildschirm ablenken lassen – sie erscheinen etwas aufgeregter und empfänglicher für neue Inputs. Mädchen hingegen wirken etwas besonnener, geduldiger und lassen sich bei der Lösung einer Aufgabe weniger leicht ablenken als Jungen. Sie hören eher der Problemstellung zu, planen ihre Lösungen voraus und sind darüber hinaus kommunikativer, wenn es um die Mitteilung von Präferenzen geht.

Es zeigt sich im Rahmen der gegenständlichen Untersuchung, dass Mädchen und Jungen unterschiedliche Lösungsstrategien einsetzen, Mädchen aber eher Erfolg haben als Jungen. Diese Beobachtungen und Erkenntnisse decken sich mit dem Bild aus der Literatur, siehe Kapitel 2.5.1, wonach Mädchen und Jungen durchaus gleich motiviert an die Bearbeitung entsprechender Aufgabenstellungen herangehen, jedoch dabei unterschiedliche Lösungsstrategien einsetzen und somit auch unterschiedliche Erfolge feiern.

11.3.5 F2.1

F2.1: Welche Möglichkeiten für einen Child-Centered-Design-Ansatz in einem österreichischen Kindergarten bestehen und welche Materialien können hierzu eingesetzt werden?

Ein Child-Centered-Design-Ansatz ist in einem österreichischen Kindergarten grundsätzlich einsetzbar. Als Materialien stehen in einem Kindergarten im Regelfall Buntstifte, Papier, kindgerechte Scheren und Klebstoff zur Verfügung. IT-Hardware zur Simulation oder zur Nutzung von Prototypen steht derzeit nicht zur Verfügung. Den Kindern sind die Tätigkeiten Zeichnen und Basteln (z.B. ausschneiden, einkleben, etc.) sehr vertraut, welche zur Entwicklung von Prototypen benötigt werden. Ebenso sind Zusammenarbeit und kooperative Bearbeitung von Aufgaben für die Kinder gewohnte Szenarien.

Durch Einsatz des Frameworks „Informatik für Kinder im Alter von 3 bis 6 Jahren“ gelingt es Kindern nach kurzer Zeit einen Bodenroboter, wie den Bee-Bot, zu programmieren (siehe Kapitel 8.6.2) und auch einzelne Teile von ihm zu benennen. Diese Fähigkeiten können genutzt werden, um Prototypen experimentell zu nutzen. Darüber hinaus können Kinder, mit ihren Erfahrungen aus den Aufgaben, die aus den entsprechenden Aktivitätsschablonen abgeleitet wurden, Prototypen zeichnen und gestalten. Das grundsätzliche Interesse der Kinder an Neuem unterstützt Aktivitäten im Bereich des Child-Centered-Designs erheblich und die Kinder verfügen über eine hohe Motivation zur Mitarbeit.

Es ist möglich, Child-Centered-Design, welches um Cooperative-Inquiry und Bonded-Design erweitert wurde, mit Kindern einzusetzen und die Rahmenbedingungen sind ähnlich zur Durchführung eines Programms zur frühen Technikbildung in Informatik (siehe Einleitung Kapitel 11.3). Basierend auf den Prototyping-Aktivitäten haben die Kinder in vier Phasen (siehe Kapitel 10.1) gemeinsam mit der Autorin Symbole gezeichnet (siehe Kapitel 10.2), ein Card-Sorting durchgeführt (siehe Kapitel 10.3) sowie einen Prototyp und seine Iteration (siehe Kapitel 10.4) evaluiert. Child-Centered-Design kann somit (unter Berücksichtigung des Alters der teilnehmenden Kinder, siehe Forschungsfrage F2.2 in Kapitel 11.3.6) mit Kindergartenkindern nicht nur zur Entwicklung neuer Produkte bzw. Benutzerschnittstellen für die Zielgruppe eingesetzt werden, sondern leistet vor allem auch einen Beitrag zur frühen Technikbildung in Informatik (siehe Kapitel 10.5).

11.3.6 F2.2

F2.2: Kann mit Hilfe eines Child-Centered-Design-Prozesses in Kombination mit Prototyping mit 3- bis 6-jährigen Kindern eine prototypische Benutzerschnittstelle einer Mikrowelt entwickelt werden, die auf einer Mini-Language basiert und vom Bee-Bot inspiriert ist?

Im Zuge eines Child-Centered-Design-Prozesses in Kombination mit Prototyping mit 5- bis 6-jährigen Kindern kann sehr gut ein Prototyp für eine benutzerfreundliche, grafische

Benutzerschnittstelle einer zielgruppengerechten Mikrowelt am PC basierend auf einer Mini-Language entwickelt werden (siehe Kapitel 10.4.2.5 und 10.4.4.5). Die Kinder dieser Altersgruppe sind motivierte Partnerinnen und Partner im kreativen Gestaltungsprozess sowie neugierige Prüferinnen und Prüfer der entstehenden Prototypen. Es zeigt sich hier deutlich, wie auch in der Literatur in Kapitel 5, dass Kinder einen wertvollen Beitrag leisten können, wenn man sie ihrem Alter und ihren Fähigkeiten entsprechend einbindet. 3- bis 4-jährige Kinder sind jedoch noch zu jung um an dem Prozess aktiv teilzunehmen, da sie unter anderem ihre Ideen noch nicht ausreichend formulieren können und mit den zum Teil etwas abstrakteren Aufgabenstellungen als gewohnt noch nicht zurechtkommen. Details dazu sind der Hypothesenprüfung H2.1 (siehe Kapitel 11.2.5) zu entnehmen.

11.3.7 F2.3

F2.3: Wie kann eine unter Einsatz eines Child-Centered-Design-Prozesses entwickelte benutzerfreundliche grafische Benutzerschnittstelle einer zielgruppengerechten Mikrowelt am PC gestaltet sein?

Die Ausgestaltung eines Prototyps einer benutzerfreundlichen grafischen Benutzerschnittstelle einer zielgruppengerechten Mikrowelt, die in Zusammenarbeit mit Vertreterinnen und Vertretern der Zielgruppe entstanden ist, ist in Kapitel 10.4.3 zu finden.

11.3.8 F2.4

F2.4: Wie muss Child-Centered-Design weiterentwickelt werden, um als Methode zur frühen Technikbildung in Informatik für Kinder im Alter von 3 bis 6 Jahren eingesetzt werden zu können?

Um Child-Centered-Design als Maßnahme zur frühen Technikbildung in Informatik nutzen zu können, ist es nach Ansicht der Autorin um Cooperative-Inquiry und Bonded-Design zu ergänzen. Es empfiehlt sich vor Einsatz des Child-Centered-Designs den teilnehmenden Kindern die Möglichkeit zu bieten, erste Programmiererfahrungen zu sammeln. Dazu eignet sich das in der gegenständlichen Arbeit entwickelte Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“. Dieses Rahmenwerk bietet den Kindern auch die Möglichkeit erste Erfahrungen für das Child-Centered-Design zu sammeln – zum Beispiel durch das Zeichnen von Prototypen eines Bodenroboters oder das Explorieren von Hard- oder Software.

Das adaptierte Child-Centered-Design in Kombination mit dem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ kann als Methode zur frühen Technikbildung in Informatik in Kindergärten eingesetzt werden, wie Kapitel 10.5 und die Prüfung der Hypothese H2.1 sowie die Beantwortung der Forschungsfrage F2.2 zeigen. Das von der Verfasserin der gegenständlichen Arbeit adaptierte Child-Centered-Design ermöglicht Kindern, andere Tätigkeiten der Informatik als Programmieren zu erleben und kennenzulernen. 3- bis 6-jährige Kinder können die unterschiedlichen Rollen wie Benutzerinnen, Benutzer, Testerinnen, Tester, Informantinnen, Informanten sowie Gestaltungspartnerinnen und -partner einnehmen. Die Rollen der Informantin, des Informanten und der Gestaltungspartnerin, des Gestaltungspartners sind für 3- bis 4-jährige Kinder jedoch sehr herausfordernd (siehe F2.2) und oft von ihnen noch nicht effektiv einnehmbar. Zusätzlich sind die Kinder aber in allen Rollen Lernende und es bestehen Lernmöglichkeiten durch die Teilnahme am Child-Centered-Design-Prozess mit seinen Aufgabenstellungen sowie durch die Einnahme unterschiedlicher Rollen.

11.4 Fazit

Der laufende österreichische und europaweite Diskurs hinsichtlich eines Mangels an hochqualifizierten Technikerinnen und Technikern am Arbeitsmarkt stellt den Ausgangspunkt der gegenständlichen Arbeit dar. Im Fokus der Arbeit stehen nicht nur die Vermittlung von Programmierwissen und die Begeisterung der Kinder für Technik, sondern auch die einhergehende Kompetenzentwicklung.

Kompetenzorientiertes Lernen wird im Kontext des lebenslangen selbstorganisierten Lernens immer wichtiger. Wie in Kapitel 2 ausgeführt, ist die Förderung kompetenzorientierten Lernens für die Zielgruppe noch rar, diese wird aber zunehmend gefordert. Diesem Umstand trägt das in der gegenständlichen Arbeit von der Autorin entwickelte Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ (siehe Kapitel 8.7) Rechnung, in dem es die Aktivitätsschablonen erlauben Aufgaben mit unterschiedlichen Tätigkeiten sowie Sozialformen abzuleiten und so die Ausprägung individueller Lernkompetenzen unterstützt wird.

Wie auch der Literatur zu entnehmen, siehe Kapitel 3.2, helfen die neuen Formen der Interaktion mit Computern bei der individuellen Kompetenzentwicklung. Die Verfasserin der gegenständlichen Arbeit hat bezüglich der Förderung von Kompetenzen in Kindergärten in Kapitel 3.2.5 einen großen Forschungsbedarf verortet. Dem folgend entwickelte sie ein kompetenzbasiertes Framework und fördert durch Aktivitätsschablonen zur Ableitung von Aufgaben in drei Leitthemenbereichen („Early Childhood Technology“, „Computational Thinking“ sowie „Child-Centered-Design“) sowohl Fachkompetenzen als auch akademische Kompetenzen. Die Clusterung der Kompetenzen erfolgt in Anlehnung an Eguchi (2007), wobei die Fachkompetenzen Informatik, Technik allgemein sowie Mathematik umfassen und die akademischen Kompetenzen Gestalten, Dokumentieren, Reflektieren, Kooperativ Arbeiten, Kreatives Denken, Entscheidungen treffen, Probleme lösen, Kommunizieren, Präsentieren und Räumliches Denken (siehe Kapitel 8.2.4).

In Kapitel 4 stellt die Autorin Ansätze und Herausforderungen vor, die bei einem Einstieg in die Programmierung für 3- bis 6-jährige Kinder zu berücksichtigen sind. Trotz intensiver Recherche konnte die Verfasserin der gegenständlichen Arbeit kein fertig ausgearbeitetes Programmangebot für die Zielgruppe der 3- bis 6-jährigen Kinder auffinden (siehe Kapitel 6), welches wissenschaftlich fundiert, kompetenzbasiert und herstellerunabhängig ist. Aus diesem Kapitel lässt sich klar ableiten, dass der Einstieg in die Programmierung über ein Tangible Device, ein angreifbares Produkt, einen zielgruppene geeigneten Roboter, erfolgen soll, da dieses für die Zielgruppe einfacher und unmittelbarer zu bedienen ist als Software, was sich auch mit den Ergebnissen der vorliegenden Forschungsarbeit deckt (siehe Kapitel 11.2 und 11.3)

Als erste Innovation der Forschungsarbeit entwickelte die Autorin ein kompetenzbasiertes und herstellerunabhängiges Framework unter dem Titel „Informatik für Kinder im Alter von 3 bis 6 Jahren“, basierend auf Erkenntnissen aus der Literatur und den empirischen Ergebnissen der Studien in den zwei Kindergärten D und E. Das Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“, siehe Kapitel 8.7, umfasst 50 verallgemeinerte Aktivitätsschablonen unterschiedlicher Schwierigkeitsgrade, Sozialformen und Inhaltsdimensionen. Dabei sind die Aktivitätsschablonen in drei Leitthemen, „Early Childhood Technology“, „Computational Thinking“ und „Child-Centered-Design“ gruppiert.

Bei der Durchführung eines Programms basierend auf dem Framework ist vor allem darauf zu achten, dass die teilnehmenden Kinder regelmäßig anwesend sind und so der Lerneffekt gewährleistet werden kann. Darüber hinaus ist für ausreichende fachliche Betreuung (Lehrpersonen für die Kinder) sowie ausreichende Betreuung der Kinder allgemein zu sorgen.

Als Forschungsmethode setzte die Autorin, basierend auf ihren Erfahrungen aus der Vorstudie, eine umfassende Video-Aufzeichnung mit anschließender qualitativer Inhaltsanalyse nach Mayring ein. Die Prüfung der Hypothesen zur Beantwortung der Forschungsfragen zeigte, dass Kinder im Alter von 3 bis 6 Jahren nach Besuch geeigneter Lehr-Lern-Einheiten, die auf dem Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ basieren, in der Lage sind, Problemstellungen desselben Schwierigkeitsgrades wie in den Lehr-Lern-Einheiten selbständig zu lösen (siehe H1.1, F1.1). Jedoch ist dabei die Voraussetzung zu beachten, dass den Kindern die ersten Schritte der Programmierung mit einem Tangible Device, im gegenständlichen Fall bei der konkreten Durchführung in den beiden Kindergärten D und E mit einem Bee-Bot Bodenroboter, vermittelt werden und erst

nach einiger Zeit Software zum Einsatz kommt. So kann der Wissenstransfer vom Bodenroboter zu korrespondierenden Softwareumgebungen bzw. Mikrowelten, für die vorliegende Arbeit unter der Bedingung der Einführung in die Bedienung der Software, erfolgreich von Statten gehen (siehe F1.2).

Am Markt stehen zielgruppengerechte Programmierumgebungen und Mikrowelten sowie Mini-Languages für 3- bis 6-jährige Kinder zur Verfügung. Kapitel 9.1 dokumentiert die Evaluierung der Hardware Usability und Kapitel 9.2 die Evaluierung der Software Usability. Die eingesetzte Hardware, der Bee-Bot der Firma TTS-Group, der mittels einer Mini-Language programmiert wird, erscheint als für die Zielgruppe der 3- bis 6-jährigen Kinder geeignet, siehe Kapitel 9.1.7. Die Software-Produkte Focus on Bee-Bot 1 bzw. 2 hingegen weisen einige Usability-Schwächen auf, siehe Kapitel 9.2.7 und Prüfung der Hypothese H1.2 in Kapitel 11.2. Das größte Manko ist die Tatsache, dass die Aufgabenstellungen und die Fehlermeldungen mit Text versehen sind und die Kinder der Zielgruppe oft noch nicht lesen können. Kinder können die eben genannten Usability-Schwächen selbst nicht identifizieren, aber in geeigneten Settings darauf hinweisen (F1.3). Es zeigte sich bei der Prüfung der Hypothese H1.2 und der Beantwortung der Forschungsfrage F1.3, dass 3- bis 6-Jährige nicht in der Lage sind, ohne vorhergehende Einführung eine softwarebasierte Programmierumgebung bzw. Mikrowelt zum Erlernen der Programmierung selbständig zu nutzen.

Die Rahmenbedingungen für eine Durchführung von Maßnahmen zur frühen Technikbildung in „Informatik für Kinder im Alter von 3 bis 6 Jahren“ sind in der Einleitung zu Kapitel 11.3 zusammengefasst. Wesentlich ist, dass für alle beteiligten Kinder ausreichend Material und Betreuungspersonal zur Verfügung stehen, damit die Motivation der Kinder hoch bleibt. Der finanzielle Aufwand hält sich für einen Start der Aktivitäten in Grenzen und ein Einstieg in die frühe Technikbildung in Informatik ist somit niederschwellig möglich.

Als zweite Innovation der vorliegenden Forschungsarbeit führte die Autorin einen Child-Centered-Design-Prozess, erweitert um Cooperative-Inquiry und Bonded-Design, mit Kindern im Alter von 5 bis 6 Jahren als Maßnahme zur frühen Technikbildung durch. Dabei wurde exemplarisch ein Prototyp einer grafischen Benutzerschnittstelle für eine Mikrowelt zum Erlernen von Programmierung für 3- bis 6-jährige Kinder entwickelt, siehe Kapitel 10.4. Es zeigte sich, siehe Prüfung der Hypothese H2.1 in Kapitel 11.2 oder auch Kapitel 10.4 und bei der Beantwortung der Forschungsfragen F2.2 und F2.3, dass 5- bis 6-jährige Kinder als Partnerinnen und Partner im Child-Centered-Design-Prozess agieren können und einen Prototyp für eine Benutzerschnittstelle, die für ihre Altersgruppe geeignet ist, mitgestalten können. 3- bis 4-jährige Kinder sind noch zu jung, als dass sie aktiv am Prozess teilnehmen und ihre Ideen ausreichend formulieren können. Sie sind dennoch in der Lage, wenn auch mit Unterstützung durch Tutorinnen und Tutoren, den von den 5- bis 6-jährigen Kindern mitgestalteten Prototyp einer Benutzerschnittstelle zu nutzen. Wie in Kapitel 10.5 dargelegt, unterstützt eine Teilnahme der Kinder am adaptierten Child-Centered-Design-Prozess beim Erreichen zahlreicher Lernergebnisse, korrespondierend zum Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“, sowie beim Kompetenzerwerb. Es werden nicht nur die Fachkompetenzen Informatik, Mathematik und Technik allgemein gefördert, sondern auch die akademischen Kompetenzen Gestalten, Dokumentieren, Reflektieren, Kooperativ Arbeiten, Kreativ Denken, Entscheidungen treffen, Probleme lösen, Kommunizieren, Präsentieren und Räumlich Denken. Die Kinder sind somit im gegenständlichen adaptierten Child-Centered-Design-Ansatz auch als Lernende zu betrachten.

11.5 Ausblick

Weiterführende Untersuchungen basierend auf den gegenständlichen Forschungsergebnissen könnten sich unter anderem mit der These beschäftigen, dass Kinder, wenn sie im Kindergartenalter bewusst mit Technik in Berührung kommen, eher einen technischen Beruf wählen. Rund 30 % der am Abschluss-Assessment dieser Forschungsarbeit teilnehmenden Kinder äußerten einen Berufswunsch im Bereich Technik,

rund 20 % wollen einen Beruf im Bereich Wissenschaft ausüben, siehe Kapitel 8.6.2.6.16. Das sind weder repräsentative Zahlen noch kann anhand dieser Äußerungen ein gesteigertes Interesse an Technik und Wissenschaft abgeleitet werden. Mit Hilfe einer Langzeitstudie könnte die Veränderung des Berufswunschs erhoben und die Auslöser für die tatsächliche Berufswahl gegenüber den früheren Wünschen dokumentiert werden. So könnte festgestellt werden, welchen Einfluss die frühe Technikbildung auf die spätere Berufswahl hat.

Als nächste Aktivität verfolgt die Autorin das Vorhaben, das in Kapitel 8.7 vorgestellte Framework „Informatik für Kinder im Alter von 3 bis 6 Jahren“ für den Einsatz weiterer Hardware zu adaptieren, um unterschiedlichen Schwerpunktsetzungen gerecht zu werden. Das vorgestellte Framework ist ausgerichtet auf einen Einsatz einfacher Bodenroboter. Mit geringen Erweiterungen könnten auch Bodenroboter mit höherer Komplexität genutzt werden.

Ein weiterführendes Advanced-Framework, das auf dem aus der gegenständlichen Arbeit aufbaut und für eine höhere Altersstufe bzw. für 5- bis 6-jährige Kinder geeignet ist, kann mit Hilfe des Pro-Bots der TTS-Group entwickelt werden. Der Pro-Bot, siehe Kapitel 4.4.2, bietet im Gegensatz zum Bee-Bot die Möglichkeit der Programmierung komplexerer Programme (z.B. Schleifen, beliebige Drehwinkel), wobei die Programmierung entweder direkt am Pro-Bot oder via PC erfolgt.

Mehr Gestaltungsmöglichkeiten und Freiheitsgrade als der Bee-Bot und der Pro-Bot bieten die Roboter von Robo Wunderkind³⁶. Das Set von Robo Wunderkind besteht aus Blöcken, ähnlich den Cubelets, siehe Kapitel 4.3.2, mit unterschiedlichen Funktionalitäten wie einer Kamera, einem Lichtsensor, einer Stromversorgung oder einem Motor. Diese Blöcke können aneinandergereiht und anschließend via App am Tablett oder Smartphone programmiert werden. Der Hersteller deutet auf der Webseite an, dass ein Einsatz für 3- bis 5-jährige Kinder möglich ist. Bei Einsatz der Robo Wunderkind Blöcke mit dem Framework der gegenständlichen Arbeit sieht die Autorin die Möglichkeit der Erweiterung des Frameworks um die Aktivitäten, den Roboter selbst zu bauen, der anschließend programmiert werden soll. Durch die Vielzahl an Blöcken können auch weitere verschiedene Schwierigkeitsgrade für die jeweiligen Altersgruppen implementiert werden.

Ein weiterer Schritt ist die Entwicklung eines Train-the-Trainer Programms, das Kindergartenpädagoginnen und -pädagogen die Möglichkeit bietet, das in dieser Arbeit vorgestellte Framework im eigenen Kindergarten einzusetzen und zu nutzen. Eine Integration derartiger Train-the-Trainer Curricula ist im Zuge der aktuellen Bestrebungen die Ausbildung der Kindergartenpädagoginnen und -pädagogen an Hochschulen zu holen³⁷ aus Sicht der Autorin wünschenswert.

Zusätzliche Forschungstätigkeiten sind beim Einsatz des Child-Centered-Design-Ansatzes als Maßnahme zur frühen Technikbildung erforderlich. In der vorliegenden Arbeit wurden erste empirische Erkenntnisse gesammelt, die es durch einen umfangreicheren Einsatz in der Praxis zu prüfen gilt. Auch die Weiterentwicklung des Child-Centered-Design-Prozesses liegt im Fokus. So könnte sich die Autorin vorstellen hierzu ein Framework mit Aktivitätsschablonen zu entwickeln und weitere Praxisbeispiele zu evaluieren, an Hand derer die Kinder an einem Child-Centered-Design-Prozess partizipieren und lernen können. Denn es muss nicht immer eine Benutzeroberfläche einer Software entwickelt werden, es könnten auch Hardware-Produkte, wie zum Beispiel ein Alltagsgegenstand, in den Fokus des Interesses rücken. Darüber hinaus ist es der Verfasserin der gegenständlichen Arbeit ein Anliegen, den adaptierten, um Bonded-Design und Cooperative Inquiry erweiterten, Child-Centered-Design-Ansatz als Maßnahme zur frühen Technikbildung in Informatik zu etablieren.

³⁶ <http://robowunderkind.com/>, 23.04.2018

³⁷ <http://karriere.nachrichten.at/aus-weiterbildung/studium/art200616.2373697>, 23.04.2018

11.6 Danksagung

Zum Abschluss möchte ich mich bei den teilnehmenden Kindergärten und hier besonders bei den Kindern, die meine Forschung und mich voll neugieriger Begeisterung begleitet und inspiriert haben, bedanken.

Mein besonderer Dank gilt Margit Pohl für die Betreuung und Begleitung während meines Dissertationsvorhabens.

Ein herzliches Danke an Werner Stöckelmayr und Michael Tesar, die bei der Dokumentation der abgehaltenen Aktivitäten, den Usability-Tests und bei der Betreuung der Kinder assistierten, sowie an das Kindergartenpersonal für die offene, interessierte und herzliche Unterstützung vor Ort.

Meiner Familie und meinen Freunden danke ich für alles und noch viel mehr.

Literaturverzeichnis

- Ackermann, E. K. (2012). Programming for the natives: What is it? What's in it for the kids. *Constructionism*.
- Addison, P. (1997). Teaching computer science without a computer. *ACM SIGCSE Bulletin*, 29(4), 30–33. Retrieved from <http://portal.acm.org/citation.cfm?id=271149>
- Agudo, J. E., Sánchez, H., & Rico, M. (2010). Playing Games on the Screen: Adapting Mouse Interaction at Early Ages. In *Proceedings of 10th IEEE International Conference on Advanced Learning Technologies 2010* (pp. 493–497). Sousse, Tunisia; 5-7 July 2010: IEEE. doi:10.1109/ICALT.2010.142
- Aldrich, J. (2002). Early childhood teacher candidates evaluate computer software for young children. *Information Technology in Childhood Education Annual*, 295–300. Retrieved from http://www.editlib.org/index.cfm/files/paper_8849.pdf?fuseaction=Reader.DownloadFullText&paper_id=8849
- Alexander, C., Digiovanni, L., & Rosen, D. (2011). Integrating Technology into Early Childhood Field Experiences. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (Vol. 2011, pp. 4476 – 4481). Nashville, Tennessee, USA March 7-10, 2011: Chesapeake, VA: AACE. Retrieved from <http://editlib.org/p/37039>
- Alexander, T., & Rackley, C. C. (2005). Integrating Information Assurance (IA) into K-5 curriculum. In *Proceedings of the 2nd annual conference on Information security curriculum development* (pp. 1–3). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1107624>
- Alimisis, D. (2008). Designing robotics-enhanced constructivist training for science and technology teachers: the TERECoP Project. In J. Luca & E. Weippl (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008* (pp. 288–293). Vienna, Austria, June 30- July 4, 2008: AACE, Chesapeake, VA. Retrieved from http://www.editlib.org/index.cfm?fuseaction=Reader.ViewFullText&paper_id=28409
- Alsumait, A., & Al-Osaimi, A. (2009). Usability heuristics evaluation for child e-learning applications. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services* (pp. 425–430). Kuala Lumpur, Malaysia, December 14–16, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1806417>
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. In *Journal of Educational Technology & Society*, 19(3).
- Angel-Fernandez, J. M., & Vincze, M. (2018). Introducing storytelling to educational robotic activities. In *Global Engineering Education Conference (EDUCON), 2018 IEEE* (pp. 608-615). IEEE.
- Angel-Fernandez, J. M., & Vincze, M. (2018b). Towards a Definition of Educational Robotics. In *Austrian Robotics Workshop 2018* (p. 37).
- Anslow, C., Markstrum, S., & Murphy-Hill, E. (2009). Evaluation and usability of programming languages and tools (plateau). In *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications* (pp. 1053–1054). Orlando, Florida, USA, October 25–29, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1640085>
- Antle, A. (2007). The CTI Framework : Informing the Design of Tangible Systems for Children. In *Proceedings of the first International Conference on Tangible and Embedded Interaction (TEI'07)* (pp. 195–202). Balton Rouge, LA, USA; 15-17 Feb., 2007: ACM.
- Antonitsch, P. K. (2013). HOW to Consider Informatics in Primary Education ? In P. Micheuz, A.Reiter, G. Brandhofer, M. Ebner, & B. Sabitzer (Eds.), *Digitale Schule Österreich* (pp. 360–371). Österreichische Computer Gesellschaft ÖCG.
- Antonitsch, P. K., Gigacher, C., Hanisch, L., & Sabitzer, B. (2013). A media-reduced approach towards informatics at primary level. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education, WiPSE '13* (pp. 118–121). Aarhus, Denmark, November 11-13, 2013: ACM Press. doi:10.1145/2532748.2532763
- Armstrong, J. (2011). Visual Motor Integration Utilizing the Internet. In M. Koehle & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 3677–3680). Nashville, Tennessee, USA: AACE.

- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670. doi:10.1016/j.robot.2015.10.008
- Baartman, L., Bastiaens, T., Kirschner, P., & van der Vleuten, C. (2007). Evaluating assessment quality in competence-based education: A qualitative comparison of two frameworks. *Educational Research Review*, 2(2), 114–129. doi:10.1016/j.edurev.2007.06.001
- Baauw, E., & Markopoulous, P. (2004). A comparison of think-aloud and post-task interview for usability testing with children. In *Proceedings of the 2004 conference on Interaction design and children: building a community* (pp. 115–116). College Park, Maryland, USA, June 1-3, 2004: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1017848>
- Bærendsen, N. K., Jessen, C., & Nielsen, J. (2009). Music-Making and Musical Comprehension with Robotic Building Blocks. In M. Chang, R. Kuo, Kinshuk, G.-D. Chen, & M. Hirose (Eds.), *Learning by Playing. Game-based Education System Design and Development Lecture Notes in Computer Science* (pp. 399–409). Berlin Heidelberg New York: Springer. doi:10.1007/978-3-642-03364-3_48
- Barendregt, W., Bekker, T. M., Börjesson, P., Eriksson, E., & Torgersson, O. (2016, June). Legitimate Participation in the Classroom Context: Adding Learning Goals to Participatory Design. In *Proceedings of the The 15th International Conference on Interaction Design and Children* (pp. 167-174). ACM.
- Barker, B. S., Grandgenett, N., Nugent, G., & Adamchuk, V. I. (2009). Scaling-up an Educational Robotics Intervention for Informal Learning Environments. In G. Siemens & C. Fulford (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009* (pp. 3231–3236). Honolulu, HI, USA: AACE.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?. *Acm Inroads*, 2(1), p. 48-54.
- Baumert, J., & Kunter, M. (2006). Stichwort: Professionelle Kompetenz von Lehrkräften. *Zeitschrift Für Erziehungswissenschaft*, 9(4), 469–520. doi:10.1007/s11618-006-0165-2
- Beals, L. M. (2010). Content creation in virtual worlds to support adolescent identity development. *New Directions for Youth Development*, (128), 45–53.
- Beals, L. M., & Bers, M. (2006). Robotic technologies: when parents put their learning ahead of their child's. *Journal of Interactive Learning Research*, 17(4), 341–366. Retrieved from http://www.editlib.org/toc/index.cfm/files/paper_19973.pdf?fuseaction=Reader.DownloadFullText&paper_id=19973
- Bekker, T., Bakker, S., Douma, I., Van Der Poel, J., & Scheltenaar, K. (2015). Teaching children digital literacy through design-based learning with digital toolkits in schools. *International Journal of Child-Computer Interaction*, 5, 29-38.
- Bell, M., & Crawford, C. (2001). Integrating Technology into the Young Child Lesson Plan. In J. Price, D. A. Willis, & N. Davis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2001* (Vol. 3, pp. 3160–3162). Orlando, Florida, USA, March 5-10, 2001: Association for the Advancement of Computing in Education (AACE). Retrieved from http://www.editlib.org/d/17355/proceeding_17355.pdf
- Bell, S., Heeler, L., & Heeler, P. (2008). A preliminary report on the use of robots with elementary school students. *Journal of Computing Sciences*, 23(4), 263–268. Retrieved from <http://portal.acm.org/citation.cfm?id=1352127>
- Ben-Ari, M. (2001). Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45–73.
- Beran, T. N., & Ramirez-Serrano, A. (2010). Do children perceive robots as alive? In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction - HRI '10* (pp. 137–138). Osaka, Japan, March 2-5 2010. doi:10.1145/1734454.1734511
- Beraza, I., Pina, A., & Demo, B. (2010). Soft & Hard ideas to improve interaction with robots for Kids & Teachers. In *Proceedings of SIMPAR 2010 Workshops Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS* (pp. 549–557). Darmstadt, Germany, November 15-16, 2010: ISBN 978-3-00-032863-3. Retrieved from <http://www.terecop.eu/SIMPARG2010/TR-TWR-2010/19-TeachingRobotics.pdf>
- Bergner, N., Hilde, K., Magenheimer, J., Kathrin, M., Ralf, R., Ulrik, S., & Carsten, S. (2017). Zieldimensionen für frühe informatische Bildung im Kindergarten und in der Grundschule. In *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*.

- Berk, L. E. (2005). *Entwicklungspsychologie* (3.Auflage ed., p. 1066). München, Boston, San Francisco, Harlow, Don Mills, Sydney, Mexico City, Madrid, Amsterdam: Pearson Studium.
- Berndt, E.-B. (2009). Integration statt Sahnehäubchen Die technologische Basis der Kulturtechniken hat sich verändert. In B. Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule", Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (pp. 230–242). Berlin, Germany, Sept. 21-24 2009: GI - Gesellschaft für Informatik.
- Berns, K., & Schmidt, D. (2010). *Programmierung mit Lego®) Mindstorms® NXT* (p. 234). Berlin Heidelberg New York: Springer Verlag, examen press.
- Bers, M. U. (2007). Positive Technological Development: Working with Computers, Children and the Internet. *MassPsych The Periodical of the Massachusetts Psychological Association*, 51(1), 5–7 & 18–19.
- Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. In *Early Childhood Research & Practice*, 12(2), n2.
- Bers, M. U., Beals, L., Chau, C., Satoh, K., & Khan, N. (2010). Virtual Worlds for Young People in a Program Context: Lessons from Four Case Studies. In M. S. Khine & I. M. Saleh (Eds.), *New Science of Learning: Cognition, Computers and Collaboration in Education* (pp. 357–383). New York: Springer Verlag.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. doi:10.1016/j.compedu.2013.10.020
- Bers, M. U., & Horn, M. S. (2010). Tangible programming in early childhood: Revisiting developmental assumptions through new technologies. In I. R. Berson & M. J. Berson (Eds.), *High-tech tots: childhood in a digital world* (pp. 49–69). Information Age Publishing. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Tangible+programming+in+Early+Childhood:+Revisiting+Developmental+Assumptions+through+New+Technologies#0>
- Bers, M. U., Rogers, C., Beals, L., Portsmore, M., Staszowski, K., Cejka, E., Carberry, A., Gravel, B., Anderson, J. & Barnett, M. (2006). Innovative Session : Early Childhood Robotics for Learning. In *Proceedings of the 7th International Conference of the Learning Sciences (ICLS 2006)* (pp. 1036–1042). Indiana, Bloomington IN, USA; June 27-July 1, 2006.
- Bierman, K. L., & Erath, S. A. (2006). Promoting Social Competence in Early Childhood : Classroom Curricula and Social Skills Coaching Programs. In K. McCartney & D. Phillips (Eds.), *The Blackwell Handbook of Early Childhood Development* (pp. 595–615). Malden, MA: Blackwell Publishing.
- Bingchen, Z., & Yanqun, W. (2010). Research on the mouse design of children. In *Proceedings of the 2nd International Conference on Networking and Digital Society (ICNDS), 2010* (pp. 113–116). Wenzhou, China, May 30-31, 2010: IEEE. doi:10.1109/ICNDS.2010.5479317
- Birbaumer, A., Tolar, M., & Wagner, I. (2006). Biographical Stories of European Women Working in ICT. In E. M. Trauth (Ed.), *Encyclopedia of Gender and Information Technology* (pp. 64–69). Hershey, London, Melbourne, Singapore: Idea Group Inc.
- Bischof, E. (2010). Informatik in der Volksschule spielerisch erLeben. In G. Brandhofer, G. Futschek, P. Micheuz, A. Reiter, & K. Schoder (Eds.), *25 Jahre Schulinformatik in Österreich. Zukunft mit Herkunft* (pp. 169–179). Wien: Österreichische Computer Gesellschaft ÖCG.
- Blackwell, A. F. (1996). Metacognitive theories of visual programming: What do we think we are doing? In *Proceedings. of IEEE Symposium on Visual Languages, 1996* (pp. 240–246). Boulder, Colorado, USA, Sep. 3-6, 1996: IEEE. doi:10.1109/VL.1996.545293
- Blikstein, P. (2013). Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13* (pp. 173–182). New York, New York, USA, June 23-27, 2013: ACM Press. doi:10.1145/2485760.2485786
- Blikstein, P., & Krannich, D. (2013). The makers' movement and FabLabs in education: experiences, technologies, and research. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13* (pp. 613–616). New York, New York, USA, June 23-27, 2013: ACM Press. doi:10.1145/2485760.2485884
- Bortz, J., & Döring, N. (2006). *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler* (4.Auflage ed., p. 700). Heidelberg: Springer Medizin Verlag Heidelberg.
- Brauner, P., Leonhardt, T., Ziefle, M., & Schroeder, U. (2010). The Effect on Tangible Artifacts, Gender and Subjective Technical Competence on Teaching Programming to Seventh Graders. In

- J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010*, (Vol. 5941, pp. 61–71). Zurich, Switzerland, January 13-15, 2010: Springer Verlag. doi:10.1007/978-3-642-11376-5
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada (Vol. 1, p. 25).
- Brito, N., & Barr, R. (2012). Influence of bilingualism on memory generalization during infancy. *Developmental Science*, 15(6), 812–6. doi:10.1111/j.1467-7687.2012.1184.x
- Brouwer, N., Muller, G. J., & Rietdijk, H. (2007). Educational Designing With MicroWorlds. *Journal of Technology and Teacher Education*, 15(4), 439–462.
- Brown, D., & Warschauer, M. (2006). From the university to the elementary classroom: Students' experiences in learning to integrate technology in instruction. *Journal of Technology and Teacher Education*, 14(3), 599–621. Retrieved from http://www.editlib.org/INDEX.CFM?fuseaction=Reader.ViewFullText&paper_id=5996
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages : A Way to Learn Programming Principles. *Education and Information Technologies*, 2(1), 65–83.
- Buldu, M. (2002). Young Children' s Computer Use : Perspectives of Early Childhood Teacher Educators. In D. A. Willis, J. Price, & N. Davis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2002* (pp. 2544–2548). Nashville, Tennessee, USA: AACE.
- Burnett Heyes, S., Zokaei, N., van der Staaij, I., Bays, P. M., & Husain, M. (2012). Development of visual working memory precision in childhood. *Developmental Science*, 15(4), 528–539. doi:10.1111/j.1467-7687.2012.01148.x
- Cacco, L., & Moro, M. (2014). When a Bee meets a Sunflower. In *Proceedings of 4th International Workshop Teaching Robotics Teaching with Robotics and 5th International Conference on Robotics in Education*, Padova, Italy (pp. 68-75).
- Cairncross, S., & Waugh, S. (2005). Involving preschoolers in design of interactive multimedia for learning: An activity-driven approach. In P. Kommers & G. Richards (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005* (pp. 1117–1124). Montreal, Canada; June 27-July 2, 2005: AACE. Retrieved from http://www.editlib.org/index.cfm/files/paper_20228.pdf?fuseaction=Reader.DownloadFullText&paper_id=20228
- Carter, J. (1999). Incorporating standards and guidelines in an approach that balances usability concerns for developers and end users. *Interacting with Computers*, 12(2), 179–206. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0953543899000119>
- Catlin, D., Kandlhofer, M., & Holmquist, S. (2018). EduRobot taxonomy: A provisional schema for classifying educational robots. In *International Conference on Robotics in Education 2018*.
- Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. In: *International Journal of Engineering Education*, 22(4), 711.
- Chambers, D., & Tromp, C. (2000). Doing IT Down Under : Uses of Information Technologies in an Australian Degree for K-6 Teachers. In D. A. Willis, J. Price, & E. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2000* (pp. 1232–1239). San Diego, California, USA: AACE. Retrieved from http://www.editlib.org/toc/index.cfm/files/paper_15808.pdf?fuseaction=Reader.DownloadFullText&paper_id=15808
- Chan, M. S., Black, J. B., Han, I., Vitale, J., Xia, Q., Subramanian, M., Du, M. & Kang, S. (2007). “ Look, it 's turning !” Factors Affecting Structural and Functional Knowledge Acquisition in an Elementary School Robotics Classroom. In C. Montgomerie & J. Seale (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* (pp. 1626–1631). Vancouver, Canada, June 25-29, 2007: AACE.
- Chang, C.-K. (2008). Usability Comparison of Pen-Based Input for Young Children on Mobile Devices. In *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 2008 (sutc 2008)* (pp. 531–536). Taichung, China; 11-13 June 2008: IEEE. doi:10.1109/SUTC.2008.8

- Chen, A., Muntz, R. R., Yuen, S., Locher, I., Park, S. I., & Srivastava, M. B. (2002). A Support Infrastructure for the Smart Kindergarten. *Pervasive Computing, IEEE*, 1(2), 49–57. doi:10.1109/MPRV.2002.1012337
- Clayton, K. L., von Hellens, L. A., & Nielsen, S. H. (2009). Gender Stereotypes Prevail in ICT; a Research Review. In *Proceedings of the 2009 ACM SIGMIS Computer Personnel Research Conference (SIGMIS-CPR'09)* (pp. 153–158). Limerick, Ireland; May 28–30, 2009: ACM.
- Clements, D. H., & Nastasi, B. K. (1992). Computers and early childhood education. *Preschool and early childhood treatment directions*, p.187-246.
- Cockburn, A., & Bryant, A. (1996). Do it This Way: Equal Opportunity Programming for Kids. In *Proceedings of the 6th Australian Conference on Computer-Human Interaction (OZCHI '96)* (pp. 246–251). Hamilton, NEW ZEALAND , Nov. 24 -27, 1996: IEEE Computer Society. doi:http://doi.ieeeecomputersociety.org/10.1109/OZCHI.1996.560017
- Cockburn, A., & Bryant, A. (1998). Cleogo: collaborative and multi-metaphor programming for kids. In *Proceedings of 3rd Asia Pacific Computer Human Interaction, 1998.* (pp. 189–194). Shonan Village Center , Japan; 15-17 July 1998: IEEE. doi:10.1109/APCHI.1998.704194
- Crawford, R. H., Wood, K. L., Fowler, M. L., & Norrell, J. L. (1994). An engineering design curriculum for the elementary grades. *Journal of Engineering Education*, 83(2), 172–181. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.8703&rep=rep1&type=pdf>
- Csink, L., & Farkas, K. (2010). Lifelong Playing Instead of Lifelong Learning Teaching Robotics without Robots and Computers. In *Proceedings of SIMPAR 2010 Workshops, International Conference on Simulation, Modeling and Programming for Autonomous Robots* (pp. 439–448). Darmstadt, Germany: ISBN 978-3-00-032863-3. Retrieved from <http://www.simpar.org/ws/sites/TR-TWR-2010/06-TeachingRobotics.pdf>
- CSTA. (2011). *CSTA K - 12 Computer Science Standards* (p. 73). New York, New York, USA. Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
- Cypher, A., & Smith, D. C. (1995). KIDSIM: END USER PROGRAMMING OF SIMULATIONS. In *Proceedings of Conference on Human Factors in Computing Systems (CHI '95 Mosaic of Creativity)* (pp. 27–34). Denver, Colorado, USA; May 7-11 1995: ACM.
- Dack, G. H., Ford, M. J., & Prejean, L. M. (2006). Technology Infusion across Pedagogy and Content Areas. In C. Crawford (Ed.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2006* (pp. 3375–3398). Orlando, Florida, USA, Mar 19, 2006: Chesapeake, VA: AACE. ISBN 978-1-880094-58-7.
- Dagiene, V., & Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In R. T. Mittermeir & M. M. Sysło (Eds.), *Proceedings of the Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008* (Vol. 5090, pp. 19–30). Torun, Poland, July 1-4, 2008: SpringerVerlag. doi:10.1007/978-3-540-69924-8
- Dagiene, V., & Futschek, G. (2010). Introducing informatics concepts through a contest. In *IIFIP Working Conference: New Developments in ICT and Education* (pp. 1–15). Universite de Picardie Jules Verne, Amiens: ISBN: 978-2-9537285-1-4. Retrieved from http://publik.tuwien.ac.at/files/PubDat_186636.pdf
- DeClue, T. (2008). Computer Science in Kindergarten? Of Course!: The Abc's of the K-12 CSTA Model Curriculum in Computer Science. *Journal of Computing Sciences*, 23(4), 257–262.
- deDiego-Cottinelli, A., & Barros, B. (2010). TRAZO: A Tool to Acquire Handwriting Skills Using Tablet-PC Devices. In *Proceedings of the 9th International Conference on Interaction Design and Children (IDC'10)* (pp. 278–281). Barcelona, Spain, June 9-12, 2010: ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=1810591>
- Demo, G. B., & Marcianó, G. (2007). Contributing to the development of Linguistic and Logical Abilities through Robotics. In I. Kalaš (Ed.), *Proceedings of the 11th European Logo Conference* (pp. 46–55). Bratislava, Aug., 19-24, 2007: Faculty of Mathematics, Physics and Informatics Comenius University Bratislava.
- Demo, G. B., Marcianò, G., & Siega, S. (2008). Concrete Programming: Using Small Robots in Primary Schools. In *Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies 2008* (pp. 301–302). Santander, Cantabria, Spain, 1-5 July 2008: IEEE. doi:10.1109/ICALT.2008.190

- Diekmann, A. (2009). *Empirische Sozialforschung - Grundlagen, Methoden, Anwendungen* (20. Aufl., p. 784). Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag.
- Dinkelaker, J., & Herrle, M. (2009). *Erziehungswissenschaftliche Videographie: Eine Einführung* (1. Aufl., p. 136). Wiesbaden: VS Verlag für Sozialwissenschaften.
- Döbeli, B. (2010). ICT im Hosensack – Informatik im Kopf? Gedanken zur ICT und Informatik in der Volksschule. In G. Brandhofer, G. Futschek, P. Micheuz, A. Reiter, & K. Schoder (Eds.), *25 Jahre Schulinformatik - Zukunft mit Herkunft* (pp. 35–44). Wien: Österreichische Computergesellschaft.
- Döbeli Honegger, B., & Hielscher, M. (2017). Vom Lehrplan zur LehrerInnenbildung-Erste Erfahrungen mit obligatorischer Informatikdidaktik für angehende Schweizer PrimarlehrerInnen. In *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*.
- Donker, A., & Reitsma, P. (2004). Usability testing with young children. In *Proceedings of the 2004 conference on Interaction design and children: building a community* (pp. 43–48). College Park, Maryland, USA, June 1-3, 2004: ACM. doi:10.1145/1017833.1017839
- Donker, A., & Reitsma, P. (2007). Drag-and-drop errors in young children's use of the mouse. *Interacting with Computers*, 19(2), 257–266. doi:10.1016/j.intcom.2006.05.008
- Donker, A., & Reitsma, P. (2007b). Young children's ability to use a computer mouse. *Computers & Education*, 48(4), 602–617. doi:10.1016/j.compedu.2005.05.001
- Dräger, T. (2008). *Gender Mainstreaming im Kindergarten*. (C. Muth, Ed.) (p. 184). Stuttgart: ibidem-Verlag.
- Eck, J., Hirschmugl-Gaisch, S., Hofmann, A., Kandlhofer, M., Rubenzer, S., & Steinbauer, G. (2013). Innovative concepts in educational robotics: Robotics projects for kindergartens in Austria. In *Austrian Robotics Workshop* (Vol. 14, p. 12).
- Eck, J., Hirschmugl-Gaisch, S., Kandlhofer, M., & Steinbauer, G. (2014). A cross-generational robotics project day: Pre-school children, pupils and grandparents learn together. In *Journal of Automation Mobile Robotics and Intelligent Systems*, 8.
- Edwards, H., & Bedyk, R. (2007). A comparison of usability evaluation methods for child participants in a school setting. In *Proceedings of the 6th international conference on Interaction design and children* (pp. 9–16). Aalborg, Denmark, June 6-8, 2007: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1297280>
- Eguchi, A. (2007). Educational Robotics for Elementary School Classroom. In R. Carlsen, K. McFerrin, J. Price, R. Weber, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2007* (pp. 2542–2549). San Antonio, Texas, USA: AACE.
- Eguchi, A. (2009). What are students learning from educational robotics ? Different Approaches to Educational Robotics. In I. Gibson, R. Weber, K. McFerrin, R. Carlsen, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2009* (pp. 3547–3554). Charleston, SC, USA: AACE.
- Eguchi, A., & Uribe, L. (2009). Integrating Educational Robotics in Elementary Curriculum. In T. Bastiaens, J. Dron & C. Xin (Ed.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009* (pp. 2128–2135). Chesapeake, VA: Chesapeake, VA: AACE.
- Eguchi, A., Zhou, C., & Shen, J. (2010). Integrating Simulation Game with Educational Robotics - RoboCupJunior 2010 CoSpace Demo Challenge. In J. Sanchez & K. Zhang (Eds.), *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2010* (Vol. 2010, pp. 1336 – 1343). Orlando, Florida, USA, Oct 18-22, 2010: Association for the Advancement of Computing in Education (AACE), Chesapeake, VA; ISBN 978-1-880094-83-9. Retrieved from <http://editlib.org/p/35741>
- Eisenberg, M., Elumeze, N., MacFerrin, M., & Buechley, L. (2009). Children's programming, reconsidered: settings, stuff, and surfaces. In ACM (Ed.), *IDC 2009 International Conference on Interaction Design and Children* (pp. 1–8). Como, Italy; June3-5, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1551790>
- Elkin, M., Sullivan, A., & Bers, M. U. (2014). Implementing a robotics curriculum in an early childhood Montessori classroom. In *Journal of Information Technology Education: Innovations in Practice*, 13, 153-169.
- Elumeze, N., & Eisenberg, M. (2005). SmartTiles: Mobility and Wireless Programmability in Children's Construction and Crafts. In *Proceedings of the 2005 IEEE International Workshop on Wireless and*

- Mobile Technologies in Education (WMTE'05)* (pp. 230–237). Tokushima, Japan, November 28–November 30: IEEE. doi:<http://doi.ieeecomputersociety.org/10.1109/WMTE.2005.58>
- Elumeze, N., Huang, Y., Meyers, J., & Eisenberg, M. (2010). "Serious" Programming Made Cuddly: A Fully End-User-Programmable Stuffed Toy. In *Proceedings of Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010* (pp. 146–150). Kaohsiung, China, 12-16 April 2010: IEEE. doi:DOI 10.1109/DIGITEL.2010.26
- Entz, S., & Galarza, S. L. (2000). *Picture This: Digital and Instant Photography Activities for Early Childhood Learning*. Thousand Oaks: Corwin Press, Inc. ISBN-0-8039-6886-8
- Europäische Kommission. (2007). *Schlüsselkompetenzen für Lebenslanges Lernen. Ein Europäischer Referenzrahmen*. (p. 12). Luxemburg: Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften.
- Europäische Kommission. (2008). *Der europäische Qualifikationsrahmen für lebenslanges Lernen* (p. 11). Luxemburg: Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften. Retrieved from http://ec.europa.eu/dgs/education_culture
- Fagan, T. (2011). Learning, Teaching and ICTs: an Early Childhood perspective. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 2954–2960). Nashville, Tennessee, USA: AACE. Retrieved from <http://www.editlib.org/p/36764>
- Federer, P., Futschek, G., & Ruegg, J. (2009). Informatics for All - Everywhere , Any Time. *CEPIS UPGRADE: The European Journal for the Informatics Professional, 20 Years of CEPIS: Informatics in Europe Today and Tomorrow, 10(4)*, 19–22. Retrieved from <http://www.upgrade-cepis.org>
- Fernaes, Y., Håkansson, M., Jacobsson, M., & Ljungblad, S. (2010). How do you Play with a Robotic Toy Animal ? A long-term study of Pleo. In *Proceedings of the Interaction Design and Children Conference 2010 (IDC'2010)* (pp. 39–48). Barcelona, Spain, June 9-12, 2010: ACM.
- Fernaes, Y., Kindborg, M., & Scholz, R. (2006). Rethinking children's programming with contextual signs. In *Proceedings of the 2006 conference on Interaction design and children* (pp. 121–128). Tampere, Finland, June 7-9, 2006: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1139105>
- Fernaes, Y., & Tholander, J. (2006). Designing for programming as joint performances among groups of children. *Interacting with Computers, 18(5)*, 1012–1031. doi:10.1016/j.intcom.2006.05.004
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. In *Computers & Education, 63*, p. 87-97.
- Flannery, L. P., Kazokoff, E., Bontá, P., Silverman, B., Bers, M. U., & Resnick, M. (2013). Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13* (pp. 1–10). New York, New York, USA, June 23-27, 2013: ACM Press. doi:10.1145/2485760.2485785
- Ford, M. J., Dack, G. H., & Prejean, L. (2006). Robotics: Implementing Problem Based Learning in Teacher Education and Field Experience. In C. M. Crawford, R. Carlsen, K. McFerrin, J. Price, R. Weber, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2006* (pp. 3410–3416). Orlando, Florida, USA: AACE. Retrieved from http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=22619
- Forišek, M., & Steinová, M. (2010). Didactic Games for Teaching Information Theory. In J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010* (Vol. 5941, pp. 86–99). Zurich, Switzerland, January 13-15, 2010: Springer Berlin Heidelberg. doi:10.1007/978-3-642-11376-5
- Frei, P., Su, V., Mikhak, B., & Ishii, H. (2000). curlybot: Designing a New Class of Computational Toys. In T. Turner, G. Szwillus, M. Czerwinski, F. Peterno, & S. Pemberton (Eds.), *Proceedings of the ACM CHI 2000 Human Factors in Computing Systems Conference* (pp. 129–136). The Hague, The Netherlands, April 1-6, 2000: ACM Press.
- Freudenberg, R. (2009). Lernen mit Etoys. In B. Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule", Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (Vol. c, pp. 86–96). Berlin, Germany, Sept. 21-24 2009: GI-Gesellschaft für Informatik e.v. (GI), Lecture Notes in Informatics, ISBN 978-3-88579-250-5.

- Fthenakis, W. E., Reichert-Garschhammer, E., & Eirich, H. (2003). *Der Bayerische Bildungs- und Erziehungsplan für Kinder in Tageseinrichtungen bis zur Einschulung Entwurf für die Erprobung. Arbeit* (p. 253). München.
- Futschek, G. (2003). Neue Impulse für den Informatikunterricht durch das universitäre Lehramt-Studium - "Unterrichtsfach Informatik und Informatikmanagement." In A. Reiter, G. Scheidl, H. Strohmer, L. Tittler, & M. Weissenböck (Eds.), *Schulinformatik in Österreich - Erfahrungen und Beispiele aus dem Unterricht* (pp. 379–382). Wien: Ueberreuter Verlag.
- Futschek, G. (2005). Explaining and Understanding LOGO Programs, a Discipline of Learning Computer Programming. In *Proceedings of the EUROLOGO 2005, "Digital Tools for Lifelong Learning"* (pp. 327–333). Warsaw, Polen: ISBN: 83-917700-8-7.
- Futschek, G. (2006). Algorithmic Thinking: The Key For Understanding Computer Science. In R. Mittermeir (Ed.), *Informatics Education – The Bridge between Using and Understanding Computers Lecture Notes in Computer Science Volume 4226/2006* (pp. 159 – 168). Springer Verlag. doi:10.1007/11915355_15
- Futschek, G. (2007). Logo-like Learning of Basic Concepts of Algorithms - Having Fun with Algorithms. In *Proceedings of the 11th European Logo Conference, Eurologo 2007 - 40 Years of Influence on Education*" (pp. 1–6). Bratislava, Slovakia, Aug 19-24, 2007: ISBN: 978-80-89186-20-4.
- Futschek, G., & Moschitz, J. (2010). Developing Algorithmic Thinking by Inventing and Playing Algorithms. In *Proceedings of Constructionism 2010: Constructionist approaches to creative learning, thinking and education* (pp. 1–10). Paris, France, Aug. 16-20, 2010: ISBN: 978-80-89186-65-5. Retrieved from http://publik.tuwien.ac.at/files/PubDat_187461.pdf
- Futschek, G., & Moschitz, J. (2011). Learning Algorithmic Thinking with Tangible Objects Eases Transition to Computer Programming. In I. Kalaš & R. Mittermeir (Eds.), *Informatics in Schools: Contributing to 21st Century Education: 5th International Conference, ISSEP 2011, Proceedings* (LNCS 7013., pp. 155 – 164). Bratislava, Slovakia, October 26-29, 2011: Springer-Verlag.
- Gallagher, M., & Allison Baurerle. (2003). Toying With Technology: Bridging the Gap Between Education and Engineering. In C. Crawford, N. Davis, J. Price, R. Weber, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2003* (pp. 3538–3541). Albuquerque, New Mexico, USA: AACE.
- Garcia-Sanjuan, G., Jaen, J., Nacher, V., & Catala, A. (2015). Design and Evaluation of a Tangible-Mediated Robot for Kindergarten Instruction. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology 2015* (pp. 1–11). Iskandar, Malaysia, November 16-19, 2015: ACM, ISBN 978-1-4503-3852-3/15/11. doi:10.1145/2832932.2832952
- Geary, D.C. (2007). Development of Mathematical Understanding. In W. Damon & R. M. Lerner (Eds.), *Handbook of Child Psychology, Volume 2* (pp. 777–810). John Wiley and Sons.
- Gelderblom, H., & Kotzé, P. (2008). Designing Technology for Young Children : What we can Learn from Theories of Cognitive Development. In R. A. Botha & C. Cilliers (Eds.), *Proceedings of the 2008 Annual Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, SAICSIT Conf. 2008* (pp. 66–75). Wilderness, South Africa, October 6-8, 2008: ACM International Conference Proceeding Series ISBN 978-1-60558-286-3.
- Gelderblom, H., & Kotzé, P. (2009). Ten Design Lessons from the Literature on Child Development and Children' s Use of Technology. In *Proceedings of the 8th International Conference on Interaction Design and Children* (pp. 52–60). Como, Italy; 3-5 June, 2009: ACM.
- Geldreich, K., Funke, A., & Hubwieser, P. (2017). Willkommen im Programmierzirkus-Ein Programmierkurs für Grundschulen. In *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*.
- Gelman, S. A., & Kalish, C. W. (2006). Conceptual Development. In W. Damon, R. M. Lerner, D. Kuhn, & R. S. Siegler (Eds.), *Handbook of Child Psychology, Volume Two: Cognition, Perception, and Language* (Sixth Edit., pp. 687–733). John Wiley and Sons.
- Giannakos, M. N., & Jaccheri, L. (2018). From players to makers: An empirical examination of factors that affect creative game development. *International Journal of Child-Computer Interaction*, 18, 27-36.

- Gibbs, D., & Roberts, S. (2003). A Jump-start in Learning? Young Children's Use of CD Rom Technology. In *Proceedings of the IFIP Working Group 3.5 Conference: Young Children and Learning Technologies* (pp. 39–47). Parramatta, New South Wales, Australia.
- Gibson, I. (2010). Changing Landscapes: Rethinking technology practice in teacher education for the 21st century from an Early Childhood Context. In D. Gibson & B. Dodge (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 1435–1442). San Diego, CA, USA, Mar 2010: Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/index.cfm/files/paper_33560.ppt?fuseaction=Reader.DownloadPresentation&paper_id=33560&paperfile_id=59624
- Giebeler, C. (2010). Zwei Geschlechter gibt es nicht... Geschlechter-Geflechte in der Pädagogik der Kindheit. *Sozial Extra. Durchblick: Perspektiven Auf Kindheit(en)*, 34(11-12), 28–36. doi:10.1007/s12054-010-0136-y
- Goh, H., & Aris, B. (2007). Using Robotics in Education: Lessons Learned and Learning Experiences. In *Proceedings of the 1st International Malaysian Educational Technology Convention* (pp. 1156–1163). Johor Bharu, Malaysia; 2-5 November 2007.
- Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International journal of child-computer interaction*, 16, 77-84.
- Gordon, M., Rivera, E., Ackermann, E., & Breazeal, C. (2015, June). Designing a relational social robot toolkit for preschool children to explore computational concepts. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 355-358). ACM.
- Green, T. R. G., & Petre, M. (1996). Usability Analysis of Visual Programming Environments: A "Cognitive Dimensions" Framework. *Journal of Visual Languages and Computing*, 7(2), 131–174. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.3584&rep=rep1&type=pdf>
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. In *Educational Researcher*, 42(1), p. 38-43.
- Gujberova, M., & Kalaš, I. (2013). Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (pp. 108-117). ACM.
- Haberman, B., Muller, O., & Averbuch, H. (2008). Multi-facet Problem Comprehension: Utilizing an Algorithmic Idea in Different Contexts. In R. T. Mittermeir & M. M. Syslo (Eds.), *Proceedings of the Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008* (pp. 180–191). Torun, Poland, July 1-4, 2008: Springer Verlag. Retrieved from <http://www.springerlink.com/index/1m37370773r74041.pdf>
- Hadjerrouit, S. (2005). Constructivism as Guiding Philosophy for Software Engineering Education. *ACM SIGCSE Bulletin*, 37(4), 45. doi:10.1145/1113847.1113875
- Hadjerrouit, S. (2009). Teaching and Learning School Informatics: A Concept-Based Pedagogical Approach. *Informatics in Education*, 8(2), 227–250. Retrieved from http://www.mii.lt/informatics_in_education/pdf/infe147.pdf
- Halford, G. S., & Andrews, G. (2006). Reasoning and Problem Solving. In W. Damon, R. M. Lerner, D. Kuhn, & R. S. Siegler (Eds.), *Handbook of Child Psychology, Volume Two: Cognition, Perception, and Language* (Sixth Edit., pp. 557–608). John Wiley & Sons, Inc.
- Han, C. C. W. (2003). Challenges of Using ICT in Hong Kong Early Childhood Settings. In *IFIP Working Group 3.5 Conference: Young Children and Learning Technologies* (pp. 49–52). Parramatta, New South Wales, Australia: Australian Computer Society, Inc.
- Hanna, L., Ridsen, K., & Alexander, K. (1997). Guidelines for usability testing with children. *Interactions*, 4(5), 9–14. doi:10.1145/264044.264045
- Hartmann, W., Bäck, G., Hajszan, M., Hartel, B., Kneidinger, L., Marek, D., Pfohl-Chalaupek & Stoll, M. (2010). *Modul für das letzte Jahr in elementaren Bildungseinrichtungen: Vertiefende Ausführungen zum bundesländerübergreifenden BildungsRahmenPlan* (p. 63). Wien: Bundesministerium für Wirtschaft, Familie und Jugend.
- Hartmann, W., Bäck, G., Hajszan, M., Hartel, B., Kneidinger, L., Stoll, M., Bair, A., Baumann, M., Grötschnig, M., Hutz, M., Minich, S., Salzmann, G., Steger, R., Strasser, G. & Thoma, M. (2009). *Bundesländerübergreifender BildungsRahmenPlan für elementare Bildungseinrichtungen in*

- Österreich. Design (p. 30). Ämter der Landesregierungen der österreichischen Bundesländer, Magistrat der Stadt Wien, Bundesministerium für Unterricht, Kunst und Kultur.
- Hartmann, W., Näf, M., & Reichert, R. (2006). *Informatikunterricht planen und durchführen* (p. 167). Berlin Heidelberg New York: Springer Verlag, examen press.
- Herper, H., & Hinz, V. (2009). Informatische Bildung im Primarbereich. In B. Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule", Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (pp. 74–85). Berlin, Germany, Sept. 21-24 2009.
- Herper, H., & Hinz, V. (2010). Klassenzimmer der Zukunft Informatische Bildung in der Volksschule. In G. Brandhofer, P. Micheuz, G. Futschek, A. Reiter, & K. Schoder (Eds.), *25 Jahre Schulinformatik in Österreich - Zukunft mit Herkunft* (pp. 349–355). Melk, Austria, September 27-29, 2010: OCG.
- Hiller, S. (2012). Best Practice in der frühkindlichen Förderung von Technik und Naturwissenschaften– Ergebnisse aus der Evaluationsstudie „MoMoTech“. In *Wissenschafts-und Technikbildung auf dem Prüfstand* (pp. 157-170). Nomos Verlagsgesellschaft mbH & Co. KG.
- Hirst, A., Johnson, J., Petre, M., Price, B., & Richards, M. (2003). What is the best programming environment/language for teaching robotics using Lego Mindstorms? *Artificial Life and Robotics*, 7(3), 124–131. doi:10.1007/s10015-003-0246-8
- Hoffmann, S., Wendlandt, K., & Wendlandt, M. (2017). Algorithmisieren im Grundschulalter. In *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*.
- Hoidn, S. (2011). Lernmodell zur Förderung von Lernkompetenzen an Hochschulen. *Zeitschrift Für Hochschulentwicklung*, 6(1), 1–18.
- Hoiseh, M., Giannakos, M. N., & Jaccheri, L. (2013). Research-derived guidelines for designing toddlers' healthcare games. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI'13* (pp. 451–456). Paris, France, April 27 - May 02, 2013: ACM Press. doi:10.1145/2468356.2468436
- Horn, M. S., Crouser, R. J., & Bers, M. U. (2012). Tangible interaction and learning: the case for a hybrid approach. *Personal and Ubiquitous Computing*, 16(4), 379–389. doi:10.1007/ s00779-011-0404-2
- Horn, M. S., & Jacob, R. J. K. (2007). Tangible Programming in the Classroom with Tern. In *Proceedings of the Conference on Computer/Human Interaction 2007 (CHI '07)* (pp. 1965–1970). San Jose, CA, USA; April 28 - May 3, 2007: ACM.
- Horn, M. S., & Jacob, R. J. K. (2007b). Designing Tangible Programming Languages for Classroom Use. In *Proceedings of the first International Conference on Tangible and Embedded Interaction 2007 (TEI 2007)* (pp. 159–162). Baton Rouge, Louisiana, USA, February 15-17, 2007: ACM. doi:10.1145/1240866.1240933
- Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. K. (2009). Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education. In *Proceedings of the 27th international conference on Human factors in computing systems* (pp. 975–984). Boston, MA, USA, April 4-9, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1518851>
- Horn, M. S., Solovey, E. T., & Jacob, R. J. K. (2008). Tangible Programming and Informal Science learning: Making TUIs Work for Museums. In *Proceedings of the 7th international conference on Interaction design and children* (pp. 194–201). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1463756>
- Huber, G. L., & Gürtler, L. (2012). *AQUAD 7 - Das Programm zur Analyse qualitativer Daten* (p. 203). Tübingen: Eigenverlag Softwarevertrieb Günter Huber.
- Hubwieser, P. (2007). *Didaktik der Informatik* (3.Auflage ed., p. 286). Berlin Heidelberg New York: Springer Verlag, examen press.
- Hubwieser, P., & Aiglstorfer, G. (2004). *Fundamente der Informatik: Ablaufmodellierung, Algorithmen und Datenstrukturen* (p. 276). München Wien: Oldenbourg Wissenschaftsverlag GmbH.
- Humbert, L., & Pasternak, A. (2010). Qualifizierung von Informatiklehrkräften. In G. Brandhofer, F. Gerald, P. Micheuz, A. Reiter, & K. Schoder (Eds.), *25 Jahre Schulinformatik in Österreich - Zukunft mit Herkunft* (pp. 154–162). Melk, Austria, September 27-29, 2010: OCG.
- Hustedt, J. T., & Barnett, W. S. (2010). Issues of Access and Program Quality. In P. Peterson, E. Baker, & B. McGaw (Eds.), *International Encyclopedia of Education, Vol.2* (Third Edit., pp. 110–119). Elsevier.

- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. In *Computers & Education*, 126, p.296-310.
- Hyun, E., & Yoon, H. (2009). Characteristics of Young Children's Utilization of a Robot during Play Time : A Case Study. In *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication* (pp. 675–680). Toyama, Japan, Sept. 27 - Oct. 2, 2009: IEEE.
- Ioannidou, A., Repenning, A., & Webb, D. C. (2009). AgentCubes: Incremental 3D end-user development. *Journal of Visual Languages & Computing*, 20(4), 236–251. doi:10.1016/j.jvlc.2009.04.001
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. In *Computers & Education*, 82, p. 263-279.
- Iwata, M., Tano, S., Fujie, Y., & Hashiyama, T. (2006). Implementation and Evaluation of Programming Environment using Real Objects for Children. In *Proceedings of the tenth International Conference on Information Visualization 2006, IV 2006* (pp. 817–822). London, UK, 5-7 July 2006: IEEE. doi:10.1109/IV.2006.61
- Jacoby, S., Gutwillig, G., Jacoby, D., Josman, N., Weiss, P., Koike, M., Itoh, Y., Kawai, N., Kitamura, Y. & Sharlin, E. (2009). PlayCubes: Monitoring constructional ability in children using a tangible user interface and a playful virtual environment. In *Proceedings of Virtual Rehabilitation International Conference, 2009* (pp. 42–49). Haifa, Israel, June 29 2009-July 2 2009: IEEE. doi:10.1109/ICVR.2009.5174203
- Janacsek, K., Fiser, J., & Nemeth, D. (2012). The best time to acquire new skills: age-related differences in implicit sequence learning across the human lifespan. *Developmental Science*, 15(4), 496–505. doi:10.1111/j.1467-7687.2012.01150.x
- Janus, M., & Brinkman, S. (2010). Evaluating Early Childhood Education and Care Programs. In P. Peterson, E. Bake, & B. McGaw (Eds.), *International Encyclopaedia of Education* (Third Edit., pp. 25–31). Elsevier.
- Jensen, C. N., Burlison, W., & Sadauskas, J. (2012). Fostering Early Literacy Skills in Children's Libraries: Opportunities for Embodied Cognition and Tangible Technologies. In *Proceedings of the 11th International Conference on Interaction Design and Children* (Vol. 1, pp. 50–59). Bremen, Germany June 12-15, 2012: AMC. doi:10.1145/2307096.2307103
- Johnson, J. (2003). Children, robotics, and education. *Artif Life Robotics*, 7, 16–21. doi:10.1007/s10015-003-0265-5
- Joly, A. V. (2007). Design and Evaluation of Interactive Cross-platform Applications for Pre-literate Children. In *Proceedings of the 6th international conference on Interaction design and children* (pp. 185–188). Aalborg, Denmark, June 6-8, 2007: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1297325>
- Joly, A. V., Pemberton, L., & Griffiths, R. (2009). Card Sorting Activities with Preschool Children. In A. Blackwell (Ed.), *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (pp. 204–213). Cambridge, United Kingdom, September 01 - 05, 2009: British Computer Society. Retrieved from <http://portal.acm.org/citation.cfm?id=1671036>
- Jordan, N. C., Kaplan, D., Nabors Oláh, L., & Locuniak, M. N. (2006). Number sense growth in kindergarten: a longitudinal investigation of children at risk for mathematics difficulties. *Child Development*, 77(1), 153–75. doi:10.1111/j.1467-8624.2006.00862.x
- Judmaier, P., & Pohl, M. (2012). Mikrowelten als Abbild der Realität im Game Based Learning. In G. Csanyi, F. Reichl, & A. Steiner (Eds.), *Proceedings of GMW 2012: Digitale Medien-Werkzeuge für exzellente Forschung und Lehre* (pp. 110–113). Vienna, Austria, Sept. 10-13, 2012: Waxmann Verlag.
- Jumail, Rambli, D. R. A., & Sulaiman, S. (2010). A Design Framework for Flashcards Based Guided Digital Storytelling. In *Proceedings of The 2nd International Conference on Computer and Automation Engineering ICCAE (2010)* (pp. 4–8). Singapore, Singapore, 26-28 Feb. 2010: IEEE. doi:10.1109/ICCAE.2010.5451834
- Jung, E. (2010). *Kompetenzerwerb: Grundlagen, Didaktik, Überprüfbarkeit* (p. 238). München: Oldenbourg Wissenschaftsverlag GmbH.
- Jung, S., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. In *Sustainability*, 10(4), p. 905.

- Kabátová, M., Kalaš, I., & Tomcsányiová, M. (2016). Programming in Slovak Primary Schools. *Olympiads in Informatics*, 10, 125-159.
- Kafai, Y. B., Peppler, K. A., Burke, Q., Moore, M., & Glosso, D. (2010). Fröbel's Forgotten Gift: Textile Construction Kits as Pathways into Play, Design and Computation. In *Proceedings of the 9th International Conference on Interaction Design and Children 2010 (IDC 10)* (pp. 214–217). Barcelona, Spain, June 9-12, 2010: ACM.
- Kahn, K. (1996). ToonTalk™ – An Animated Programming Environment for Children. *Journal of Visual Languages & Computing*, 7(2), 197–217. doi:10.1006/jvlc.1996.0011
- Kahn, K. (1996b). Drawings on Napkins, Video-Game Animation, and Other Ways to Program Computers. *Communications of the ACM*, 39(8), 49–59.
- Kahn, K. (2014). TOONTALK REBORN - Re-implementing and re-conceptualising ToonTalk for the Web. In *Constructionism and Creativity conference* (pp. 1–8). Vienna Austria, August 19-23, 2014.
- Kalaš, I. (2010). Recognizing the potential of ICT in early childhood education – Analytical Survey UNESCO Institute for Information Technologies in Education. Moskau. IITE/INS/AS/2010/01
- Kalaš, I. (2013). Integration of ICT in early childhood education. In *X World Conference On Computer Education. Torun, Poland*.
- Kalaš, I. (2014). Programming at pre-primary and primary levels: the pipeline can start that early. *KEYCIT 2014: Key competencies in informatics and ICT*, 7, 29.
- Kalaš, I. (2016). On the road to sustainable primary programming. *Proceedings of the Constructionism in Action: Constructionism*, 184-191.
- Kalaš, I., & Winczer, M. (2008). Informatics as a Contribution to the Modern Constructivist Education. In R. Mittermeir & M. M. Syslo (Eds.), *Proceedings of the Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008* (pp. 229–240). Torun, Poland, July 1-4, 2008: Springer Verlag. doi:10.1007/978-3-540-69924-8
- Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., & Huber, P. (2016). Artificial intelligence and computer science in education: From kindergarten to university. In *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.
- Kazakoff, E., & Bers, M. (2012). Programming in a Robotics Context in the Kindergarten Classroom: The Impact on Sequencing Skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371–391.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, 41(4), 245–255. doi:10.1007/s10643-012-0554-5
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83–137. Retrieved from <http://dl.acm.org/citation.cfm?id=1089734>
- Kindborg, M., & McGee, K. (2007). Visual programming with analogical representations: Inspirations from a semiotic analysis of comics. *Journal of Visual Languages & Computing*, 18(2), 99–125. doi:10.1016/j.jvlc.2007.01.002
- Kindborg, M., & Sökjer, P. (2007). How Preschool Children Used a Behaviour-Based Programming Tool. In *Proceedings of IDC 2007 International Conference on Interaction Design and Children* (pp. 149–152). Aalborg, Denmark, June 6-8, 2007: ACM.
- Kinnula, M., Iivari, N., Isomursu, M., & Kinnula, H. (2018). Socializers, achievers or both? Value-based roles of children in technology design projects. *International Journal of Child-Computer Interaction*.
- Knobelsdorf, M. (2009). Unterricht im Wandel? – Das Potenzial der Informatik. In B. Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule", Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (pp. 218–229). Berlin, Germany, Sept. 21-24 2009: GI - Gesellschaft für Informatik.
- Knoll, B., & Ratzler, B. (2010). *Gender Studies in den Ingenieurwissenschaften* (1. Auflage., p. 220). Wien: Facultas Verlags- und Buchhandels AG.
- Kolczyk, E. (2008). Algorithm–Fundamental Concept in Preparing Informatics Teachers. In R. T. Mittermeir & M. M. Syslo (Eds.), *Proceedings of the third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008* (pp. 265–271). Torun, Poland July 1-4, 2008: Springer. doi:10.1007/978-3-540-69924-8_24

- Kucuk, S., & Sisman, B. (2017). Behavioral patterns of elementary students and teachers in one-to-one robotics instruction. In *Computers & Education*, 111, p31-43.
- Kussmaul, C., & Jack, R. (2006). User Interface Prototyping: Tips And Techniques. *Journal of Computing Sciences in Colleges*, 21(6), 188–190.
- Landoni, M., Rubegni, E., & Nicol, E. (2018). A comparative study into how pupils can play different roles in co-design activities. *International Journal of Child-Computer Interaction*, 17, p. 28-38
- Lara, S., & Whittier, D. (2004). Preparing Teachers, Student Teachers, and Schools for the 21st Century: Review of Recent Reports in the U.S. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004* (pp. 3626–3631). Lugano, Switzerland: AACE. Retrieved from http://www.editlib.org/d/12037/proceeding_12037.pdf
- Larkin, T., & Budny, D. (2005). Learning Styles in the Classroom: Approaches to Enhance Student Motivation and Learning. In *Proceedings of the 6th International Conference on Information Technology Based Higher Education and Training 2005 (IThET'05)* (pp. F4D–1 – F4D–8). Juan Dolio, Dominican Republic; July 7-9, 2005: IEEE.
- Lau, K. W., Tan, H. K., Erwin, B. T., & Petrovic, P. (1999). Creative learning in school with LEGO (R) programmable robotics products. In *Proceedings of the 29th Annual Frontiers in Education Conference, 1999. FIE'99*. (Vol. 2, pp. 12D4–26 – 12D4–31). San Juan, Puerto Rico, November 10 - 13, 1999: IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=841676
- Leclerc, M. . (2006). Benefits and Changes Coming from Technology in an Elementary French School located in a Minority Setting: A Case Study. In C. M. Crawford, R. Carlsen, K. McFerrin, J. Price, W. Roberta, & D. Anna (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2006* (pp. 1677–1684). Orlando, Florida, USA: AACE. Retrieved from http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=22303
- Lemons, M. A., & Parzinger, M. (2007). Gender Schemas: A Cognitive Explanation of Discrimination of Women in Technology. *Journal of Business and Psychology*, 22(1), 91–98. doi:10.1007/s10869-007-9050-0
- Leung, L. (2006). Learners as users, and users as learners. In *Proceedings of the 7th International Conference on Information Technology Based Higher Education and Training 2006* (pp. 541–544). Ultimo, Australia; 10-13 July 2006: IEEE. doi:10.1109/ITHET.2006.339810
- Levy, S. T., & Mioduser, D. (2008). Does it “want” or “was it programmed to...”? Kindergarten children’s explanations of an autonomous robot’s adaptive functioning. *International Journal of Technology and Design Education*, 18, 337–359. doi:10.1007/s10798-007-9032-6
- Li, D., Han, I., Kang, S., Lu, C., & Black, J. (2009). Enhance Understanding of Science Concepts with Technology-based Learning Tools (Programming and Hand-held Device) in a Lego Robotics Elementary After-School Classroom. In G. Siemens & C. Fulford (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009* (pp. 2957–2961). Honolulu, HI, USA; June 22-26, 2009: AACE. Retrieved from <http://www.editlib.org/p/31899>
- Li, W. (2001). Constructivist Learning Systems: A New Paradigm. In *Proceedings of IEEE International Conference on Advanced Learning Technologies 2001* (pp. 433–434). Madison, WI , USA; Aug. 06 - 08, 2001: IEEE.
- Liebal, J., & Exner, M. (2011). *Usability for Kids: Ein Handbuch zur ergonomischen Gestaltung von Software und Websites für Kinder (Schriften zur Medienproduktion)* (1. Auflage., p. 243). Wiesbaden: Vieweg+Teubner Verlag | Springer Fachmedien.
- Lim, C., & Tay, L. Y. (2003). Information and Communication Technologies (ICT) in an elementary school: students’ engagement in higher order thinking. *Journal of Educational Multimedia and Hypermedia*, 12(4), 425–451. Retrieved from <http://www.questia.com/googleScholar.qst?docId=5006473169>
- Lonigan, C. J., Allan, N. P., & Lerner, M. D. (2011). Assessment of Preschool Early Literacy Skills: Linking Children’s Educational Needs with Empirically Supported Instructional Activities. *Psychology in the Schools*, 48(5), 488–501. doi:10.1002/pits.20569
- Lord, S. M., Chen, J. C., Nottis, K., Stefanou, C., Prince, M., & Stolk, J. (2010). Role of Faculty in Promoting Lifelong Learning: Characterizing Classroom Environments. In *Proceedings of IEEE EDUCON Education Engineering 2010 – The Future of Global Learning Engineering Education* (pp. 381–386). Madrid, Spain; April 14-16, 2010: IEEE.

- Lorenz, J. H. (2005). Zentrale Lernstandsmessung in der Primarstufe—Vergleichsarbeiten Klasse 4 (VERA) in sieben Bundesländern. *Zdm*, 37(4), 317–323. doi:10.1007/BF02655818
- Louca, L. (2004). Programming environments for young learners: a comparison of their characteristics and students' use. In *Proceedings of the 2004 conference on Interaction design and children: building a community* (pp. 129–130). College Park, Maryland, USA, June 1-3, 2004: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1017855>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. In *Computers in Human Behavior*, 41, p. 51-61.
- Ma, Y., Prejean, L., & Williams, D. (2011). Technology-Enhanced Thematic Unit in a Pedagogical Laboratory. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 3697–3701). Nashville, Tennessee, USA: AACE. Retrieved from <http://www.editlib.org/p/36900>
- MacFarlane, S., Sim, G., & Horton, M. (2005). Assessing Usability and Fun in Educational Software. In *Proceedings of the 2005 conference on Interaction design and children (IDC 2005)* (pp. 103–109). Boulder, Colorado, USA; June 8-10, 2005: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1109554>
- Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. In *Proceedings of the 39th ACM Technical Symposium on Computer Science Education* (pp. 367–371). Portland, Oregon, USA, March 12-15, 2008: ACM.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), 1–15. doi:10.1145/1868358.1868363
- Manches, A., & Plowman, L. (2015). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*, 1–11. doi:10.1111/bjet.12355
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014, June). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference* (pp. 1-29). ACM.
- Mano, A., & Campos, J. C. (2005). A study on usability criteria regarding interfaces for children. In *Workshop on Child Computer Interaction: Methodological Research at INTERACT 2005 Conference* (pp. 1–4). Rome, Italy, Sept. 13, 2005. Retrieved from <http://repositorium.sdum.uminho.pt/handle/1822/3505>
- Mansor, E. I., De Angeli, A., & De Bruijn, O. (2008). Little fingers on the tabletop: A usability evaluation in the kindergarten. In *Proceedings of 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008*. (pp. 93–96). Amsterdam, Niederlande, 1-3 Oct. 2008: IEEE. doi:10.1109/TABLETOP.2008.4660190
- Markopoulos, P., Read, J., MacFarlane, S., & Höysniemi, J. (2008). *Evaluating Children's Interactive Products: Principles and Practices for Interaction Designers* (p. 369). Burlington, Massachusetts: Morgan Kaufmann Publishers, Inc.
- Markopoulos, P., Verschoor, Y., Ijsselsteijn, W., & de Ruyter, B. (2008b). Testing interactive products with the robot intervention method. In *Proceedings of the 7th international conference on Interaction design and children* (pp. 117–120). Chicago, IL, USA, July 11-13, 2008: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1463735>
- Markovac, V., & Rogulja, N. (2009). Key ICT Competences of Kindergarten Teachers. In Š. Vladimir, B.-T. Ljubica, & Z. Hubinkova (Eds.), *Proceedings of the 3rd International Conference on Advances and Systematic Research ECNSI – 2009; 8th Special Focus Symposium on ICESKS: Information, Communication and Economic Sciences in the Knowledge Society* (pp. 72–77). Zadar, Croatia; Nov. 12-14, 2009: Faculty of Teacher Education of the University of Zagreb.
- Marshall, P. (2007). Do tangible interfaces enhance learning? In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 163–170). Baton Rouge, Louisiana, USA, February 15-17, 2007: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1227004>
- Martin, S. S., SeEVERS, R. L., & Crawford, C. M. (2002). Computer use within learning environments: Early childhood case study. In D. A. Willis, J. Price, & N. Davis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2002* (pp. 2552–2553). Nashville, Tennessee, USA: AACE. Retrieved from http://www.editlib.org/toc/index.cfm/files/paper_17749.pdf?fuseaction=Reader.DownloadFullText&paper_id=17749

- Matthews, D., Lieven, E., & Tomasello, M. (2007). How toddlers and preschoolers learn to uniquely identify referents for others: a training study. *Child Development*, 78(6), 1744–1759. doi:10.1111/j.1467-8624.2007.01098.x
- Mattila, J., & Väättänen, A. (2006). UbiPlay: an interactive playground and visual programming tools for children. In *Proceedings of the 2006 conference on Interaction design and children* (pp. 129–136). Tampere, Finland, June 7-9, 2006: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1139106>
- Mayerová, K. (2012). Pilot Activities: LEGO WeDo at Primary School. In *Proceedings of the 3rd International Workshop Teaching Robotics, Teaching with Robotics* (pp. 32–39). Riva del Garda, Trento, Italy, April 20, 2012: ISBN: 978-88-95872-05-6.
- Mayring, P. (2002). *Einführung in die Qualitative Sozialforschung* (5. Auflage., p. 170). Weinheim und Basel: Beltz Verlag.
- Mayring, P. (2010). *Qualitative Inhaltsanalyse - Grundlagen und Techniken* (11. aktual., p. 144). Weinheim und Basel: Beltz Verlag.
- Mazuka, R., Jincho, N., & Oishi, H. (2009). Development of Executive Control and Language Processing. *Language and Linguistics Compass*, 3(1), 59–89. doi:10.1111/j.1749-818X.2008.00102.x
- Mazzone, E., & Read, J. (2005). Not just bits of paper: design sessions with children. *Journal of the Department of Computing, University of Central Lancashire, Preston, UK*, (4), 1–5. Retrieved from http://www.chici.org/references/not_just_bits_of_paper.pdf
- Mazzone, E., Xu, D., & Read, J. (2007). Design in Evaluation: Reflections on Designing for Children's Technology. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 2* (Vol. 2, pp. 153–156). Lancaster, UK, September 3-7, 2007: British Computer Society. Retrieved from <http://portal.acm.org/citation.cfm?id=1531447>
- McCabe, P. C., & Altamura, M. (2011). Empirically valid strategies to improve social and emotional competence of preschool children. *Psychology in the Schools*, 48(5), 513–540. doi:10.1002/pits.20570
- McDermott, P., Leigh, N. M., & Perry, M. A. (2002). Development and validation of the preschool learning behaviors scale. *Psychology in the Schools*, 39(4), 353–365. doi:10.1002/pits.10036
- McKenney, S., & Voogt, J. (2010). Technology and young children: How 4-7 year olds perceive their own use of computers. *Computers in Human Behavior*, 26(4), 656–664. doi:10.1016/j.chb.2010.01.002
- McKnight, L., & Fitton, D. (2010). Touch-screen technology for children: giving the right instructions and getting the right responses. In *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 238–241). Barcelona, Spain, June 9-12, 2010: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1810580>
- McKnight, L., & Read, J. C. (2009). Designing the 'record' button: using children's understanding of icons to inform the design of a musical interface. In *Proceedings of the 8th International Conference on Interaction Design and Children* (pp. 258–261). Como, Italy; June 3-5, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1551847>
- McNerney, T. (2004). From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing*, 8(5), 326–337. doi:10.1007/s00779-004-0295-6
- Meadows, S. (1998). Children learning to think: learning from others? Vygotskian theory and educational psychology. *Educational and Child Psychology*, 15(2), 6–13.
- Micheuz, P. (2009). Zahlen, Daten und Fakten zum Informatikunterricht an den Gymnasien Österreichs. In Bernhard Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule"*, *Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (pp. 230–242). Berlin, Germany, Sept. 21-24 2009: GI - Gesellschaft für Informatik.
- Misirli, A., & Komis, V. (2014). Robotics and programming concepts in early childhood education: A conceptual framework for designing educational scenarios. In *Research on e-Learning and ICT in Education* (pp. 99-118). Springer, New York, NY.
- Mittermeir, R. T. (2010). Informatikunterricht zur Vermittlung allgemeiner Bildungswerte. In *25 Jahre Schulinformatik in Österreich - Zukunft mit Herkunft* (pp. 54–73). Melk, Austria, September 27-29, 2010: OCG Österreichische Computer Gesellschaft.

- Mittermeir, R. T. (2013). Algorithmics for Preschoolers – A Contradiction? *Creative Education*, 04(09), 557–562. doi:10.4236/ce.2013.49081
- Mittermeir, R. T., Bischof, E., & Hodnigg, K. (2010). Showing Core-Concepts of Informatics to Kids and Their Teachers. In J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010* (Vol. 5941, pp. 143–154). Zurich, Switzerland, January 13-15, 2010: Springer Berlin Heidelberg. doi:10.1007/978-3-642-11376-5
- Mokhtar, R., Lehat, M. L., Basir, N. M., & Sokman, Y. (2015). A Tangible Game-Based Approach for Introducing Programming to Elementary School Students. *International Journal of Information and Education Technology*, 5(4), 246–249. doi:10.7763/IJNET.2015.V5.510
- Montemayor, J. (2001). Physical programming: software you can touch. In *Proceedings of Human factors in computing systems - CHI '01* (pp. 81–82). New York, New York, USA; 31 March - 5 April 2001: ACM Press. doi:10.1145/634067.634118
- Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W., & D'Amour, A. (2002). Physical Programming: Designing Tools for Children to Create Physical Interactive Environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI '02)* (pp. 299–306). Minneapolis, Minnesota, USA; 20-25 April 2002: ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=503430>
- Morgado, L., Bulas Cruz, M. G., & Kahn, K. (2003). Taking programming into kindergartens: Exploratory research activities using ToonTalk. In *Proceedings of Eurologo'2003 - Re-inventing technology on education* (pp. 178–189). Porto, Portugal; August 27-30, 2003. Retrieved from <http://hal.archives-ouvertes.fr/hal-00190134/>
- Morgado, L., Cruz, M., & Kahn, K. (2006). Radia Perlman—A pioneer of young children computer programming. *Current developments in technology-assisted education. Proceedings of m-ICTE*, 1903-1908.
- Morgado, L., & Kahn, K. (2008). Towards a specification of the ToonTalk language. *Journal of Visual Languages & Computing*, 19(5), 574–597. doi:10.1016/j.jvlc.2007.10.002
- Mulley, U., & Zuliani, B. (2013). Digitales Kompetenzmodell für die Volksschule. In P. Micheuz, A. Reiter, G. Brandhofer, M. Ebner, & B. Sabitzer (Eds.), *Proceedings of bmukk eEducation Sommertagung, Digitale Schule Österreich*. Klagenfurt am Wörthersee, Austria, Aug. 26-28, 2013.
- Neset, V., & Large, A. (2004). Children in the information technology design process: A review of theories and their applications. *Library & Information Science Research*, 26(2), 140–161. doi:10.1016/j.lisr.2003.12.002
- Neumann, K., Fischer, H. E., & Kauertz, A. (2010). From PISA to Educational Standards : The Impact of Large-Scale Assessments on Science Education in Germany. *International Journal of Science and Mathematics Education*, 8(3), 545–563.
- Newcombe, N. S., & Huttenlocher, J. (2006). Development of Spatial Cognition. In W. Damon, R. M. Lerner, D. Kuhn, & R. S. Siegler (Eds.), *Handbook of Child Psychology, Volume 2, Cognition, Perception, and Language* (Sixth Edit., pp. 734–776). John Wiley & Sons, Inc.
- Nielsen, C., Overgaard, M., Pedersen, M., Stage, J. & Stenild, S. (2006). It's worth the hassle! The added value of evaluating the usability of mobile systems in the field. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles (NordiCHI '06)* (pp. 272–280). Oslo, Norway; 14-18 October 2006: ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=1182504>
- Nielsen, J. (1992). Finding Usability Problems through Heuristic Evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '92)* (pp. 373–380). Monterey, CA, USA; May 3-7, 1992: ACM. doi:10.1145/142750.142834
- Nielsen, J. (1993). *Usability Engineering* (p. 362). Amsterdam Boston Heidelberg London New York Oxford Paris San Diego San Francisco Singapore Sydney Tokyo: Morgan Kaufmann Publishers, Inc.
- Nielsen, J., & Molich, R. (1990). Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90* (pp. 249–256). Seattle, Washington, United States; April 1990: ACM Press. doi:10.1145/97243.97281
- Nievergelt, J. (1999). "Roboter programmieren" - ein Kinderspiel: Bewegt sich auch etwas in der Allgemeinbildung? *Informatik-Spektrum*, 22(5), 364–375. Retrieved from <http://www.springerlink.com/index/AV7X7WEPB5KHT0YG.pdf>

- Nikiforidou, Z., & Pange, J. (2010). Teachers' evaluation of preschool educational software: the case of probabilistic thinking. *Procedia - Social and Behavioral Sciences*, 9, 537–541. doi:10.1016/j.sbspro.2010.12.193
- Nir-Gal, O., & Klein, P. S. (2004). Computers for cognitive development in early childhood: The teacher's role in the computer learning environment. *Information Technology in Childhood Education Annual*, 97–119. Retrieved from http://www.editlib.org/d/12896/article_12896.pdf
- Noma, H., Sasamoto, H., Itoh, Y., Kitamura, Y., Kishino, F., & Tetsutani, N. (2003). Computer learning system for pre-school-age children based on a haptized model railway. In *Proceedings of First Conference on Creating, Connecting and Collaborating Through Computing, 2003, C5 2003*. (pp. 118–119). Kyoto, Japan; 31 Jan. 2003: IEEE. doi:10.1109/C5.2003.1222343
- NQR. (2011). *Nationaler Qualifikationsrahmen Österreich: Erläuterungen zu den EQR-Deskriptoren*. Wien. Retrieved from <http://www.lebenslanges-lernen.at/nqr>
- Ntuli, E., & Kyei-Blankson, L. (2011). Teacher Criteria for Evaluating and Selecting Developmentally Appropriate Computer Software. *Journal of Educational Multimedia and Hypermedia*, 20(2), 179–193.
- Okita, S. Y., Schwartz, D. L., Shibata, T., & Tokuda, H. (2005). Exploring young children's attributions through entertainment robots. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, 2005* (pp. 390–395). Nashville, TN, USA; 13-15 Aug. 2005: IEEE. doi:10.1109/ROMAN.2005.1513810
- Oschuetz, L., Wessolek, D., & Sattler, W. (2010). Constructing with movement: kinematics. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction* (pp. 257–260). Cambridge, Massachusetts, USA, January 25–27, 2010: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1709938>
- Pane, J. F. (1997). A Programming System for Children that is Designed for Usability. In *7th Workshop on Empirical Studies of Programmers: Graduate Student Workshop* (pp. 1–8). Alexandria, VA, October 24, 1997.
- Pane, J. F. (2001). Human-Centered Design of a Programming System for Children. In IEEE (Ed.), *Children's Programming Odyssey, HCC 2001 IEEE Symposia on Human-Centric Computing Languages and Environments* (pp. 1–2). Stresa, Italy, September 5-7, 2001: Citeseer. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.6965&rep=rep1&type=pdf>
- Pane, J. F., & Myers, B. A. (2002). The impact of human-centered features on the usability of a programming system for children. In *CHI'02 extended abstracts on Human factors in computing systems* (pp. 684–685). Minneapolis, Minnesota, USA, April 20-25, 2002: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=506545>
- Pane, J. F., Myers, B. A., & Miller, L. B. (2002). Using HCI Techniques to Design a More Usable Programming System. In *Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)* (pp. 198–206). Arlington, VA, September 3-6, 2002: IEEE. doi:10.1109/HCC.2002.1046372
- Papastergiou, M. (2006). Training prospective computer engineers in computer science education and educational technology: Students' responses and lessons learned. In E. Pearson & P. Bohman (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006* (pp. 1605–1612). Orlando, Florida, USA: AACE. Retrieved from <http://www.editlib.org/p/23470>
- Papert, S. (1970). Teaching Children Thinking. In *World Conference on Computer Education IFIPS* (Vol. 5, pp. 353–365). Amsterdam.
- Papert, S. (1993). *Mindstorms: Children, Computers and Powerful Ideas* (2. Auflage, p. 223). New York, NY, USA: Basic Books, A Member of The Perseus Books Group.
- Papert, S. (2005). You Can't Think About Thinking Without Thinking About Thinking About Something. *Contemporary Issues in Technology and Teacher Education*, 5(3/4), 366–367.
- Pardo, S., Vetere, F., & Howard, S. (2006). Teachers' involvement in usability testing with children. In *Proceedings of the 2006 conference on Interaction design and children* (pp. 89–92). Tampere, Finland, June 7-9, 2006: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1139097>
- Patrick, H., Mantzicopoulos, P., & Samarapungavan, A. (2009). Motivation for Learning Science in Kindergarten: Is There a Gender Gap and Does Integrated Inquiry and Literacy Instruction Make a Difference. *Journal of Research in Science Teaching*, 46(2), 166–191. doi:10.1002/tea.20276

- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. & Paterson, J. (2007). A survey of Literature on the Teaching of Introductory Programming. *ACM SIGCSE Bulletin*, 39(4), 204–223. doi:10.1145/1345375.1345441
- Pekárová, J. (2008). Using a Programmable Toy at Preschool Age: Why and How? In *First International Conference on Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2008* (pp. 112–121). Venice, Italy; Nov. 3-7, 2008. Retrieved from <http://www.terecop.eu/downloads/simbar2008/pekarova.pdf>
- Pohl, M., & Lanzenberger, M. (2008). How to Explain the Underrepresentation of Women in Computer Science Studies. In A. Briggie, K. Waelbers, & P. A. E. Brey (Eds.), *Current Issues in Computing and Philosophy Volume 175 Frontiers in Artificial Intelligence and Applications* (pp. 184–193). Amsterdam: IOS Press.
- Pohl, M., Rester, M., & Judmaier, P. (2009). Interactive Game Based Learning: Advantages and Disadvantages. In C. Stephanidis (Ed.), *Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Part III: Applications and Services* (pp. 92–101). San Diego, CA, USA, July 19-24, 2009: Springer Verlag.
- Pohl, M., & Schmalzl, M., (2010). Guidelines zur Evaluierung von Lernprogrammen. Unveröffentlichtes Manuskript. Technische Universität Wien: Institut für Gestaltungs- und Wirkungsforschung.
- Prenzel, M., Reiss, K., & Hasselhorn, M. (2009). Förderung der Kompetenzen von Kindern und Jugendlichen. In J. Milberg (Ed.), *Förderung des Nachwuchses in Technik und Gesellschaft* (acatech DI., pp. 15–60). Berlin Heidelberg: Springer Verlag. doi:10.007/978-3-642-01123-8
- Rader, C., Brand, C., & Lewis, C. (1997). Degrees of comprehension: children's understanding of a visual programming environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '97)* (pp. 351–358). Atlanta, GA, USA; 22-27 March 1997: ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=258793>
- Radu, I., & MacIntyre, B. (2009). Augmented-Reality Scratch: A Tangible Programming Environment for Children. In *ACM CHI 09 Workshop on Tangibles for Children* (pp. 210–213). Boston, MA, USA, April 4-9, 2009: ACM, ISBN: 978-1-60558-395-2. doi:10.1145/1551788.1551831
- Raffle, H. (2006). Kinesthetic media: touch, toys & interactive materials. In *Proceedings of the 33rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2006)* (pp. 3–5). Boston, MA, 30 July - 3 August 2006: ACM.
- Raffle, H. S., Parkes, A. J., & Ishii, H. (2004). Topobo: A Constructive Assembly System with Kinetic Memory. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2004)* (Vol. 6, pp. 647–654). Vienna, Austria; April 24-29, 2004.
- Ramírez-Benavides, K., & Guerrero, L. A. (2015). MODEBOTS: Environment for Programming Robots for Children Between the Ages of 4 and 6. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje*, 10(3), 152–159. doi:10.1109/RITA.2015.2452692
- Ramsenthaler, C. (2013). Was ist "Qualitative Inhaltsanalyse?" In M. W. Schnell, C. Schulz, H. Kolbe, & C. Dunger (Eds.), *Der Patient am Lebensende - Eine Qualitative Inhaltsanalyse* (pp. 23–42). Wiesbaden: Springer Fachmedien. doi:10.1007/978-3-531-19660-2_2
- Rapeepisarn, K., Wong, K., Fung, C., & Depickere, A. (2006). Similarities and differences between „learn through play“ and „edutainment“. In *Proceedings of the 3rd Australasian conference on Interactive entertainment* (pp. 28–32). Murdoch University, Perth, Australia; 4-6 December 2006. Retrieved from <http://dl.acm.org/citation.cfm?id=1231899>
- Razak, F. H. A., Hafit, H., Sedi, N., Zubaidi, N. 'Atiqah, & Haron, H. (2010). Usability Testing with Children: Laboratory vs Field Studies. In *Proceedings of International Conference on User Science Engineering (i-USER), 2010* (pp. 104–109). Shah Alam, Selangor, Malaysia, 13-15 Dec. 2010: IEEE.
- Read, J. C., MacFarlane, S., & Casey, C. (2002). Pens Behaving Badly – Usability of Pens and Graphics Tablets for Text Entry with Children. In *Proceedings of the 15th annual symposium of User Interface Software & Technology 2002 (UIST '02 companion)* (pp. 21–22). Paris, France; 27-30 Oct. 2002: ACM.
- Read, J., Gregory, P., MacFarlane, S., McManus, B., Gray, P., & Patel, R. (2002). An Investigation of Participatory Design with Children – Informant, Balanced and Facilitated Design. In P. Markopoulos & M. Kersten-Tsikalkina (Eds.), *Proceedings of Interaction Design and Children Workshop 2002* (pp. 53–64). Eindhoven, The Netherlands, August 28-29 2002: Citeseer. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.1324&rep=rep1&type=pdf>

- Read, J., MacFarlane, S., & Casey, C. (2006). Endurability, Engagement and Expectations: Measuring Children's Fun. In *Proceedings of the 2006 Conference on Interaction Design and Children* (pp. 81–88). Tampere, Finland — June 07 - 09, 2006: Shaker Publishing.
- Read, J. C., & Markopoulos, P. (2013). Child–computer interaction. *International Journal of Child-Computer Interaction*, 1(1), 2-6.
- Reed, P., Holdaway, K., Isensee, S., Buie, E., Fox, J., Williams, J., & Lund, A. (1999). User interface guidelines and standards: progress, issues, and prospects. *Interacting with Computers*, 12(2), 119–142. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0953543899000089>
- Reichert, R., Nievergelt, J., & Hartmann, W. (2005). *Programmieren mit Kara: Ein spielerischer Zugang zur Informatik* (2.Auflage ed., p. 152). Berlin Heidelberg New York: Springer Verlag, examen press.
- Repenning, A. (1994). Bending Icons: Syntactic and Semantic Transformation of Icons. In *Proceedings of the 1994 IEEE Symposium on Visual Languages* (pp. 296–303). St. Louis, MO, USA: IEEE.
- Repenning, A. (2000). AgentSheets: an Interactive Simulation Environment with End-User Programmable Agents. In *Proceedings of Interaction 2000* (pp. 1–8). Tokyo, Japan, 2000.
- Repenning, A., & Citrin, W. (1993). Agentsheets: Applying grid-based spatial reasoning to human-computer interaction. In *Proceedings of 1993 IEEE Symposium on Visual Languages* (pp. 77–82). Bergen, Norway: IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=269581
- Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. In *Proceedings of the 6th Creativity & Cognition Conference* (pp. 1–6). Washington, DC USA, June 13-15, 2007: ACM.
- Resnick, M. (2014). Give P's A Chance: Projects, Peers, Passion, Play. In *Constructionism and Creativity conference* (pp. 1–8). Vienna Austria, August 19-23, 2014.
- Resnick, M., Flanagan, M., Kelleher, C., MacLaurin, M., Ohshima, X., Perlin, K., & Torres, R. (2009). Growing Up Programming: Democratizing the Creation of Dynamic, Interactive Media. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)* (pp. 3293–3296). Boston, MA, USA, April 4-9, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1520472>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60–67. doi:10.1145/1592761.1592779
- Rester, M., Pohl, M., Banovic, S., Hinum, K., Miksch, S., Ohmann, S., & Popow, C. (2006). Zum hohen Stellenwert einer Usability-Untersuchung in der Evaluierung einer interaktiven Informationsvisualisierung. In *Information nutzbar machen, Zusammenfassung der Beiträge zum Usability Day IV* (pp. 143–147). Dornbirn, Austria, June 9, 2006: Pabst Science Publisher.
- Rice, M. L., & Slaten, M. (2010). Raising Elementary Students' Expectations and Achievement with Technology Integration. In *Proceedings of the International Society for Technology in Education Conference 2010*. Denver, Colorado, USA, June 27-30, 2010. Retrieved from http://center.uoregon.edu/conferences/ISTE/2010/handout_uploads/KEY_49939442/Rice_UsingTechnologyinKindergarten_RP.pdf
- Rick, J., Fleck, R., Yuill, N., Francois, P., Fields, B., & Carr, A. (2010). Lo-fi prototyping to design interactive-tabletop applications for children. In *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 138–146). Barcelona, Spain, June 9-12, 2010: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1810559>
- Rivera, H., Galarza, S., Entz, S., & Tharp, R. G. (2002). Technology and Pedagogy in Early Childhood Education: Guidance from Cultural-Historical-Activity Theory and Developmentally Appropriate Instruction. *Information Technology in Childhood Education Annual*, 181–204. Retrieved from http://www.edilib.org/index.cfm/files/paper_10765.pdf?fuseaction=Reader.DownloadFullText&paper_id=10765
- Rode, J. A., Stringer, M., Toye, E. F., Simpson, A. R., & Blackwell, A. F. (2003). Curriculum-Focused Design. In *Proceedings of the 2003 conference on Interaction design and children* (pp. 119–126). Preston, UK: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=953553>

- Romeike, R. (2011). Kreativität im Informatikunterricht. In S. Schubert & A. Schwill (Eds.), *Didaktik der Informatik* (2. Auflage, pp. 355–375). Heidelberg: Spektrum Akademischer Verlag. doi:10.1007/978-3-8274-2653-6
- Rosen, D. (2006). Emerging Research Agenda for Technology and Young Children. In C. M. Crawford, R. Carlsen, K. McFerrin, J. Price, R. Weber, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2006* (pp. 4287–4291). Chesapeake, VA: Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=22782
- Rosen, D., & Jaruszewicz, C. (2007). Developmentally Appropriate Practice and Technology Literacy of Young Learners. In R. Carlsen, K. McFerrin, J. Price, R. Weber, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2007* (pp. 2754–2757). San Antonio, Texas, USA: AACE.
- Rubenzler, S., Richter, G., & Hofmann, A. (2015). Development of a multi-grade curriculum: project “READY”. In *International Conference on Robotics in Education 2015*.
- Ruble, D. N., Martin, C. L., & Berenbaum, S. A. (2007). Gender Development. In W. Damon & R. M. Lerner (Eds.), *Handbook of Child Psychology, Volume 3: Social, Emotional, and Personality Development* (pp. 858–932). John Wiley & Sons, Inc. doi:10.1002/9780470147658.chpsy0314
- Ryokai, K., Lee, M. J., & Breitbart, J. M. (2009). Multimodal Programming Environment for Kids: A “Thought Bubble” Interface for the Pleo Robotic Character. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems* (pp. 4483–4488). Boston, MA, USA, April 4-9, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1520687>
- Sağkes, M., Trundle, K. C., Bell, R. L., & O’Connell, A. A. (2011). The influence of early science experience in kindergarten on children’s immediate and later science achievement: Evidence from the early childhood longitudinal study. *Journal of Research in Science Teaching*, 48(2), 217–235. doi:10.1002/tea.20395
- Saffer, D. (2010). *designing for interaction - Creating Innovative Applications and Devices* (2nd editio., p. 223). Berkeley, CA: New Riders - Voices that Matter.
- Samarapungavan, A., Mantzicopoulos, P., & Patrick, H. (2008). Learning science through inquiry in kindergarten. *Science Education*, 92(5), 868–908. doi:10.1002/sce.20275
- Sarodnick, F., & Brau, H. (2011). *Methoden der Usability Evaluation. Wissenschaftliche Grundlagen und praktische Anwendung* (2. überarb., p. 278). Bern: Verlag Hans Huber: Wirtschaftspsychologie in Anwendung.
- Scaife, M., & Rogers, Y. (1998). Kids as Informants: Telling us what we didn’t know or confirming what we knew already? In A. Druin (Ed.), *The design of children’s technology* (pp. 27–50). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Schachtner, C., Roth, C., & Frankl, G. (2005). Mediales Lernen im Kindergarten: Ein mediendidaktisch-pädagogisches Konzept zum Einsatz des Computers im Vorschulalter. *Medienimpulse: Pisa Und Medienkompetenz*, 53(september 2005), 39–52.
- Scharf, F., Winkler, T., Hahn, C., Wolters, C., & Herczeg, M. (2012). Tangicons 3.0: An Educational Non-Competitive Collaborative Game. In *Proceedings of the 11th International Conference on Interaction Design and Children - IDC ’12* (pp. 144–151). Bremen, Germany, June 12-15, 2012: ACM Press. doi:10.1145/2307096.2307113
- Scharf, F., Winkler, T., & Herczeg, M. (2008). Tangicons: Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class. In *Proceedings of the 7th International Conference on Interaction Design and Children, IDC 2008* (pp. 242–249). Chicago, IL, USA, June 11-13, 2008: ACM.
- Schepers, S., Dreessen, K., & Zaman, B. (2018). Rethinking children’s roles in Participatory Design: The child as a process designer. *International journal of child-computer interaction*, 16, 47-54.
- Schmidt, D., Baran, E., Thompson, A., Mishra, P., Koehler, M., & Shin, T. (2009). Examining Preservice Teachers’ Development of Technological Pedagogical Content Knowledge in an Introductory Instructional Technology Course. In I. Gibson, R. Weber, K. McFerrin, R. Carlsen, & D. A. Willis (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2009* (pp. 4145–4151). Charleston, SC, USA: AACE.
- Schubert, S., & Schwill, A. (2011). *Didaktik der Informatik* (2. Auflage., p. 418). Heidelberg: Spektrum Akademischer Verlag. doi:10.1007/978-3-8274-2653-6

- Schweikardt, E., & Gross, M. D. (2006). roBlocks: A Robotic Construction Kit for Mathematics and Science Education. In *Proceedings of the 8th international conference on Multimodal interfaces - ICMI '06* (pp. 72–75). Banff, Canada; November 2-4, 2006: ACM Press. doi:10.1145/1180995.1181010
- Schweikardt, E., & Gross, M. D. (2008). Learning about Complexity with Modular Robots. In *Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning* (pp. 116–123). Banff, Canada; November 17-19, 2008: IEEE. doi:http://doi.ieeecomputersociety.org/10.1109/DIGITEL.2008.49
- Secrist, C., de Koeyer, I., Bell, H., & Fogel, A. (2002). Combining Digital Video Technology and Narrative Methods for Understanding Infant Development. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, 3(2), 18. Retrieved from <http://www.qualitative-research.net/index.php/fqs/article/viewArticle/863>
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the ninth annual international ACM conference on International computing education research - ICER '13* (pp. 59–66). San Diego in San Diego, California USA, August 12-14, 2013: ACM Press. doi:10.1145/2493394.2493403
- Seitinger, S., Sylvan, E., Zuckerman, O., Popovic, M., & Zuckerman, O. (2006). A New Playground Experience : Going Digital ? In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI 2006)* (pp. 303–308). Montreal, Quebec, Canada; April 22-27, 2006.
- Sheehan, R. (2003). Children's Perception of Computer Programming as an Aid to Designing Programming Environments. In *Proceedings of the 2003 conference on Interaction design and children* (pp. 75–83). Preston, UK: ACM. doi:10.1145/953536.953548
- Sheehan, R., & Read, J. (2009). A 2D Physics Simulator as a Design Tool for a Children's Programming Environment. In G. Siemens & C. Fulford (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009* (pp. 1647–1652). Honolulu, HI, USA; June 22-26, 2009: AACE. Retrieved from http://www.editlib.org/d/31698/_31698.pdf
- Silva, A. F., Barros, R. P., Azevedo, S. O., Silva, A., & Gonçalves, L. M. G. (2008). Diagnostic robotic agent in the roboeduc environment for educational robotics. In *Proceedings of IEEE Latin American Robotic Symposium 2008* (pp. 131–136). Natal, Rio Grande do Norte , Brazil; 29-30 Oct. 2008: IEEE. doi:10.1109/LARS.2008.16
- Sim, G., MacFarlane, S., & Read, J. (2006). All work and no play: Measuring fun, usability, and learning in software for children. *Computers & Education*, 46(3), 235–248. doi:10.1016/j.compedu.2005.11.021
- Sluis-Thiescheffer, R. J. W. (2007). How to optimize early design methods with children? In *Proceedings of the 6th international conference on Interaction design and children* (pp. 201–204). Aalborg, Denmark, June 6-8, 2007: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1297329>
- Smith, A. C. (2006). Tangible Cubes as Programming Objects. In *Proceedings of the 16th International Conference on Artificial Reality and Telexistence–Workshops, 2006*. (pp. 157–161). Hangzhou, China, Nov. 29 2006-Dec. 1 2006: IEEE Computer Society. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4089231
- Smith, A. C. (2009). Simple Tangible Language Elements for Young Children. In *Proceedings of the 8th International Conference on Interaction Design and Children* (pp. 288–289). Como, Italy; June3-5, 2009: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1551860>
- Smith, D. C. (1994). KIDSIM: Programming Agents Without a Programming Language. *Communications of the ACM*, 37(7), 54–67. doi:10.1145/176789.176795
- Smith, D. C., Cypher, A., & Schmucker, K. (1996). Making Programming easier for Children. *Interactions*, 3(5), 59–67. doi:10.1145/234757.234764
- Smith, D. C., Cypher, A., & Tesler, L. (2000). Novice Programming Comes of Age. *Communications of the ACM*, 43(3), 75–81.
- Smith, K. (2017). *New materials for teaching computational thinking in early childhood education* (Master thesis, Massachusetts Institute of Technology).
- Solga, H., & Pfahl, L. (2009). Doing Gender im technisch-naturwissenschaftlichen Bereich. In J. Milberg (Ed.), *Förderung des Nachwuchses in Technik und Gesellschaft* (acatech DI., pp. 155–218). Berlin Heidelberg: Springer Verlag. doi:10.1007/978-3-642-01123-8

- Spannagel, C., & Zendler, A. (2008). Prozessorientierte Informatikdidaktik : Welche Inhaltskonzepte sind relevant? *Notes on Educational Informatics - Section A: Concepts and Techniques*, 4(2), 19–32.
- Squires, D., & Preece, J. (1999). Predicting quality in educational software: Evaluating for learning, usability and the synergy between them. *Interacting with Computers*, 11(5), 467–483. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0953543898000630>
- Steinová, M., & Sišková, J. (2010). Programming Camps: Letting Children Discover the Computer Science. In J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010* (Vol. 5941, pp. 170–181). Zurich, Switzerland, January 13-15, 2010: SpringerVerlag. doi:10.1007/978-3-642-11376-5
- Stöckelmayr, K., Tesar, M., & Hofmann, A. (2011). Kindergarten Children Programming Robots: A First Attempt. In R. Stelzer & K. Jafarmadar (Eds.), *Proceedings of the 2nd International Conference on Robotics in Education, RiE 2011* (pp. 185 – 192). Vienna, Austria; September, 15-16, 2011, ISBN: 978-3-200-02273-7. Retrieved from http://www.robotchallenge.org/fileadmin/user_upload/_temp_/RiE/Proceedings/51.pdf
- Stockinger, J., & Futschek, G. (2016) Informatisches Denken in der Primarstufe: die Wiener Zauberschule der Informatik (WIZIK). *GI-Edition Lecture Notes in Informatics*, P-259 (2016), S. 1165-1168.
- Stoilescu, D., & Beckford, C. (2009). Constructivist and Critical Theory Aspects in Computer Science Education. In T. Bastiaens, J. Dron, & C. Xin (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009* (pp. 3168–3175). Vancouver, Canada: AACE.
- Straker, L. M., Coleman, J., Skoss, R., Maslen, B. A., Burgess-Limerick, R., & Pollock, C. M. (2008). A comparison of posture and muscle activity during tablet computer, desktop computer and paper use by young children. *Ergonomics*, 51(4), 540–555. doi:10.1080/00140130701711000
- Strawhacker, A., & Bers, M. U. (2014). “I want my robot to look for food”: Comparing Kindergartner’s programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, 25(3), 293–319. doi:10.1007/s10798-014-9287-7
- Strawhacker, A., Lee, M., Caine, C., & Bers, M. (2015). ScratchJr Demo: A coding language for Kindergarten. In *Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15* (pp. 414–417). Medford, MA, USA, June 21-25, 2015: ACM Press. doi:10.1145/2771839.2771867
- Streim, N., Horwitz, J., Pica, T., & Abel, J. (1995). Hands-on Science for Grades K-5: An Industry, School, University Collaboration. In *Proceedings of 1995 Frontiers in Education Conference* (pp. 13–19). Atlanta, GA, USA: IEEE.
- Stringer, P. (1998). One night Vygotsky had a dream: “children learning to think...” and implications for educational psychologists. *Educational and Child Psychology*, 15(2), 14–20.
- Su Kuen Seong, D. (2006). Usability Guidelines for Designing Mobile Learning Portals. In *Proceedings of the 3rd international conference on Mobile technology, applications & systems - Mobility '06* (pp. 25–32). New York, New York, USA: ACM Press. doi:10.1145/1292331.1292359
- Sugar, W. (2005). Instructional Technologist as a Coach: Impact of a Situated Professional Development Program on Teachers’ Technology Use. *Journal of Technology and Teacher Education*, 13(4), 547–571. Retrieved from http://cmappublic2.ihmc.us/rid=1140645857093_1763269249_2991/tech_coach.pdf
- Sullivan, A., & Bers, M. U. (2015). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 1–18. doi:10.1007/s10798-015-9304-5
- Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO Robot Demo: Engaging Young Children in Programming and Engineering. In *Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15* (pp. 418–421). Medford, MA, USA, June 21-25, 2015: ACM Press. doi:10.1145/2771839.2771868
- Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The Wheels on the Bot go Round and Round: Robotics Curriculum in Pre-Kindergarten. *Journal of Information Technology Education: Innovations in Practice*, 12, 203–219. Retrieved from <http://www.jite.org/documents/Vol12/JITEv12IIPp203-219Sullivan1257.pdf>

- Sullivan, F. R., & Moriarty, M. A. (2009). Robotics and Discovery Learning: Pedagogical Beliefs, Teacher Practice, and Technology Integration. *Journal of Technology and Teacher Education*, 17(1), 109–142. Retrieved from <http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ831338>
- Sylla, C. (2010). Get the picture? Evaluating Interfaces through Children's Drawings. In *Proceedings of the 28th of the international conference* (pp. 4381–4386). Atlanta, GA, USA, April 12–13, 2010: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1754157>
- Thorpe, K., Cloney, D., & Tayler, C. (2010). Rethinking Early Childhood Education and Care: Implications for Research and Evaluation. In P. Peterson, E. Baker, & B. McGaw (Eds.), *International Encyclopedia of Education* (Third Edit., pp. 144–150). Elsevier.
- Tomitsch, M., Grechenig, T., Kappel, K., & Költringer, T. (2006). Experiences from designing a Tangible Musical Toy for Children. In *Proceedings of the 2006 Conference on Interaction Design and Children* (pp. 169–170). Tampere, Finland, June 7-9, 2006: ACM. doi:10.1145/1139073.1139078
- Tsan, J., Boyer, K. E., & Lynch, C. F. (2016). How Early Does the CS Gender Gap Emerge? A Study of Collaborative Problem Solving in 5th Grade Computer Science. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16* (pp. 388–393). Memphis, Tennessee, USA, March 2-5, 2016: ACM Press. doi:10.1145/2839509.2844605
- Tsukamoto, H., Takemura, Y., Nagumo, H., Ikeda, I., Monden, A., & Matsumoto, K. I. (2015). Programming education for primary school children using a textual programming language. In *Frontiers in Education Conference (FIE), 2015 IEEE* (pp. 1-7). IEEE.
- Turja, L. (2008). *UPDATE Understanding & Providing a Developmental Approach to Technology Education: D2.2 Description of best pedagogical practises*. Projekt co-funded by the European Commission within the Sixth Framework Programme (p. 39). University of Jyväskylä, Department of Educational Sciences/Early Childhood Education
- Turja, L. (2009). *UPDATE Understanding & Providing a Developmental Approach to Technology Education: D2.4.Suggestions for teacher training focusing on early childhood education. Framework*. Projekt co-funded by the European Commission within the Sixth Framework Programme (p. 31). University of Jyväskylä, Department of Educational Sciences/Early Childhood Education
- Turja, L. (2009b). *UPDATE Understanding & Providing a Developmental Approach to Technology Education: D2.5. Suggestions for early childhood education curriculum development for enhanced technology education*. Projekt co-funded by the European Commission within the Sixth Framework Programme. (p. 11). University of Jyväskylä, Department of Educational Sciences/Early Childhood Education.
- Valente, A. (2004). Exploring Theoretical Computer Science Using Paper Toys (for kids). In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)* (pp. 301–305). Joensuu, Finland, 30 August - 1 September 2004: IEEE.
- van Doorn, F., Stappers, P. J., & Gielen, M. (2013). Design Research by Proxy: using Children as Researchers to gain Contextual Knowledge about User Experience. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (pp. 2883–2892). Paris, France, April 27 - May 02, 2013: ACM Press. doi:10.1145/2470654.2481399
- van Kesteren, I. E. H., Bekker, M. M., Vermeeren, A. P. O. S., & Lloyd, P. A. (2003). Assessing Usability Evaluation Methods on their Effectiveness to Elicit Verbal Comments from Children Subjects. In *Proceedings of the 2003 conference on Interaction design and children* (pp. 41–49). Preston, UK: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=953544>
- Verenikina, I. (2010). Vygotsky in Twenty-First-Century Research. In C. Montgomerie, J. Herrington, & B. Hunter (Eds.), *Proceedings of the World Confernece on Educational Multimedia, Hypermedia und Telecommunications (EDMEDIA) 2010* (pp. 16–25). Toronto, Canada; June 29 - July 2, 2010: ACE.
- Verenikina, I., Herrington, J., Peterson, R., & Mantei, J. (2010). Computers and Play in Early Childhood: Affordances and Limitations. *Journal of Interactive Learning Research*, 21(1), 139–159. Retrieved from <http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ890755>
- Verhaegh, J., Fontijn, W., & Jacobs, A. (2008). On the Benefits of Tangible Interfaces for Educational Games. In *Proceedings of the Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, 2008* (pp. 141–145). Banff, BC , Canada, Nov. 17-19, 2008: IEEE. doi:10.1109/DIGITEL.2008.37

- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. doi:10.1007/s10639-015-9412-6
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. (V. John-Steiner, E. Souberman, M. Cole, & S. Scribner, Eds.) (Revised ed., p. 176). Harvard University Press.
- Wahab, N. A., Osman, A., & Ismail, M. H. (2010). Engaging Children to Science Subject: A Heuristic Evaluation of Mobile Learning Prototype. In *Proceedings of the second International Conference on Computer Engineering and Applications 2010* (pp. 513–516). Bali, Indonesia; 19-21 March 2010: IEEE. doi:10.1109/ICCEA.2010.248
- Walsh, G., Foss, E., Yip, J., & Druin, A. (2013). FACIT PD: A Framework for Analysis and Creation of Intergenerational Techniques for Participatory Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (pp. 2893–2902). Paris, France, April 27 - May 02, 2013: ACM Press. doi:10.1145/2470654.2481400
- Wang, D., Zhang, C., & Wang, H. (2011). T-Maze: A Tangible Programming Tool for Children. In *Proceedings of the 10th International Conference on Interaction Design and Children* (pp. 127–135). Ann Arbor, USA, June, 20-23, 2011: ACM. Retrieved from http://allenzhang.org/ChengZhang_Files/TMaze IDC2011.pdf
- Wang, F., Kinzie, M., McGuire, P., & Pan, E. (2008). Technology-Enhanced, Problem-Based Inquiry Learning in Early Childhood Education: A Theoretical Basis. In K. McFerrin, R. Weber, R. Carlsen, & D. A. Willis (Eds.), *Proceedings of the Society for Information Technology & Teacher Education International Conference 2008* (Vol. 19, pp. 2266–2271). Las Vegas, Nevada, USA: AACE. Retrieved from http://www.editlib.org/d/27545/proceeding_27545.pdf
- Wang, L.-C., & Chen, M.-P. (2010). Learning Programming Concepts through Game Design: A PCT Perspective. In *Proceedings of the third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning 2010* (pp. 219–221). Kaohsiung, China ;12-16 April 2010: IEEE. doi:10.1109/DIGITEL.2010.47
- Weigend, M. (2008). To Have or to Be? Possessing Data Versus Being in a State – Two Different Intuitive Concepts Used in Informatics. In R. T. Mittermeir & M. M. Sysło (Eds.), *Informatics Education-Supporting Computational Thinking* (Vol. 5090, pp. 151–160). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-69924-8
- Weigend, M. (2009). Algorithmik in der Grundschule. In B. Koerber (Ed.), *Zukunft braucht Herkunft: 25 Jahre "INFOS - Informatik und Schule", Proceedings of the 13th GI-Fachtagung Informatik und Schule, INFOS 2009* (pp. 97–108). Berlin, Germany, Sept. 21-24 2009.
- Weigend, M. (2010). Mission to Mars – A Study on Naming and Referring. In J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Proceedings of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010* (Vol. 5941, pp. 182–193). Zurich, Switzerland, January 13-15, 2010: Springer Verlag. doi:10.1007/978-3-642-11376-5_17
- Wentzel, W. (2008). *Schriftenreihe des Kompetenzzentrums Technik. Diversity. Chancengleichheit. Heft 7: „Ich will das und das ist mein Weg!“ – Junge Frauen auf dem Weg zum Technikberuf. Qualitative Interviews mit ehemaligen Girls' Day-Teilnehmerinnen in Ausbildung und Studium* (p. 139). Bielefeld: Kompetenzzentrum Technik-Diversity-Chancengleichheit e.V.
- Wilgosh, L. (2002). Examining Gender Images, Expectations, and Competence as Perceived Impediments to Personal, Academic and Career Development. *International Journal for the Advancement of Counselling*, 24(4), 239–260. doi:10.1023/A:1023320813239
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717–3725. doi:10.1098/rsta.2008.0118
- Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE '15* (pp. 55–60). London, United Kingdom, November 09-11, 2015: ACM Press. doi:10.1145/2818314.2818340
- Wood, E., Willoughby, T., Schmidt, A., Porter, L., Specht, J., & Gilbert, J. (2004). Assessing The Use of Input Devices for Teachers and Children in Early Childhood Education Programs. *Information Technology in Childhood Education Annual*, 261–280. Retrieved from http://www.editlib.org/INDEX.CFM?fuseaction=Reader.ViewFullText&paper_id=5732

- Woods, S., Davis, M., Dautenhahn, K., & Schulz, J. (2005). Can Robots be Used as a Vehicle for the Projection of Socially Sensitive Issues? Exploring Children's Attitudes Towards Robots Through Stories. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, 2005* (pp. 384–389). Nashville, TN, USA; 13-15 Aug. 2005: IEEE. doi:10.1109/ROMAN.2005.1513809
- Wook, T. S. M. T., Ashaari, N. S., Noor, S. F. M., Zin, N. A. H. M., & Jaafar, A. (2011). Framework of Visual Interface for Children's Digital Library. In *Proceedings of International Conference on Electrical Engineering and Informatics (ICEEI), 2011* (pp. 1–4). Bandung, Indonesia, 17-19 July 2011: IEEE. doi:10.1109/ICEEI.2011.6021585
- Wright, T. (2006). PatternProgrammer: Yet Another Rule-Based Programming Environment for Children. In W. Piekarski (Ed.), *Proceedings of the 7th Australasian User interface conference- Volume 50* (pp. 91–96). Hobart, Tasmania, Australia; January 16-19, 2006: Australian Computer Society, Inc. Retrieved from <http://portal.acm.org/citation.cfm?id=1151770>
- Wu, C. C., Tseng, I., & Huang, S. L. (2008). Visualization of Program Behaviors: Physical Robots Versus Robot Simulators. In R. T. Mittermeir & M. M. Syslo (Eds.), *Proceedings of the Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008* (Vol. 5090, pp. 53–62). Torun, Poland, July 1-4, 2008: Springer Verlag. doi:10.1007/978-3-540-69924-8
- Wyeth, P. (2006). Ethnography in the Kindergarten: Examining Children's Play Experiences. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 1225–1228). Montréal, Québec, Canada, April 22-27, 2006: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1124956>
- Wyeth, P., & Purchase, H. C. (2000). Programming without a computer: A new interface for children under eight. In *Proceedings of the First Australasian User Interface Conference, 2000. AUIC 2000.* (pp. 141–148). Canberra, ACT, Australia, Jan., 31 - Feb. 3, 2000: IEEE. doi:10.1109/AUIC.2000.822080
- Wyeth, P., & Purchase, H. C. (2002). Designing Technology for Children: Moving from the Computer into the Physical World with Electronic Blocks. *Information Technology in Childhood Education Annual*, 219–244. Retrieved from http://www.editlib.org/INDEX.CFM?fuseaction=Reader.ViewFullText&paper_id=10767
- Wyeth, P., & Purchase, H. C. (2002b). Tangible Programming Elements for Young Children. In *CHI '02 extended abstracts on Human factors in computing systems - CHI'02* (pp. 774–775). Minneapolis, Minnesota, USA, April 20-25, 2002: ACM Press. doi:10.1145/506558.506591
- Wyeth, P., & Wyeth, G. (2001). Electronic Blocks: Tangible Programming Elements for Preschoolers. In M. Hirose (Ed.), *Proceedings of the 8th IFIP TC13 Conference on Human-Computer Interaction* (pp. 496–503). Tokyo, Japan, July 9-13, 2001: ISBN: 4 274 90451. Retrieved from <http://archive.itee.uq.edu.au/~peta/WyethInteract.pdf>
- Xiajian, C., Danli, W., & Hongan, W. (2011). Design and Implementation of a Graphical Programming Tool for Children. In *Proceedings of the IEEE International Conference on Computer Science and Automation Engineering 2011* (pp. 572–576). Shanghai, China; 10-12 June 2011: IEEE. doi:10.1109/CSAE.2011.5952915
- Xu, D. (2007). Design and Evaluation of Tangible Interfaces for Primary School Children. In *Proceedings of the 6th international conference on Interaction design and children* (pp. 209–212). Aalborg, Denmark, June 6-8, 2007: ACM. doi:10.1145/1297277.1297331
- Xu, D., Mazzone, E., & MacFarlane, S. (2006). In Search for Evaluation Methods for Children's Tangible Technology. In *Proceedings of the 2006 Conference on Interaction Design and Children* (pp. 171–172). Tampere, Finland, June 7-9, 2006: ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1139079>
- Xu, D., Read, J. C., Mazzone, E., & Brown, M. (2007). Designing and Testing a Tangible Interface Prototype. In *Proceedings of the 6th international conference on Interaction Design and Children* (Vol. 2, pp. 25–28). Aalborg, Denmark, June 6-8, 2007: ACM. doi:10.1145/1297277.1297282
- Xu, D., Read, J., Mazzone, E., MacFarlane, S., & Brown, M. (2007). Evaluation of Tangible User Interfaces (TUIs) for and with Children – Methods and Challenges. In J. Jacko (Ed.), *Human-Computer Interaction, Part II, HCII2007, LNCS4551* (pp. 1008–1017). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-73107-8_111
- Yatim, M. H. M., Shariff, A. S., Fathil, N. S., Zaki, N. A. A., Ibharam, L. F., Abdullah, A., & Supal, S. N. (2011). Managing Icon Presentation in Children Software: Interpretation from Children

- Perspectives. In *Proceedings of the 7th International Conference on Information Technology in Asia (CITA 11), 2011* (pp. 1–5). Kuching, Sarawak, Malaysia, 12-13 July 2011: IEEE.
doi:10.1109/CITA.2011.5998953
- Yip, J. C., Foss, E., Bonsignore, E., Guha, M. L., Norooz, L., Rhodes, E., McNally, B., Papadatos, P. Golub, E. & Druin, A. (2013). Children Initiating and Leading Cooperative Inquiry Sessions. In *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13* (pp. 293–296). New York, New York, USA, June 23-27, 2013: ACM Press.
doi:10.1145/2485760.2485796
- Yost, N. (2003). Computers, Kids, and Crayons : A Comparative Study of Emergent Writing Behaviours. In *Proceedings of the IFIP Working Group 3.5 Conference: Young Children and Learning Technologies* (pp. 107–112). Parramatta, New South Wales, Australia; July 2003: Australian Computer Society, Inc.

Abkürzungsverzeichnis

z.B. zum Beispiel

AS Abschluss-Assessment

HWT Hardware Test synonym zu Hardware Usability Test oder Evaluierung Hardware Usability

SWT Software Test synonym zu Software Usability Test oder Evaluierung Software Usability

Abbildungsverzeichnis

Abbildung 1: Aufbau der Arbeit.....	6
Abbildung 2: Die Hauptaktivitäten in der Technikbildung, eigene Darstellung nach (Turja, 2009b, S. 7)	23
Abbildung 3: Informatische Bildung, eigene Darstellung nach (Hubwieser, 2007, S. 49)	30
Abbildung 4: Das PLU-Modell, eigene Darstellung nach (Markopoulos et al., 2008, S. 29) ..	63
Abbildung 5: Interaktionsanteil am Gestaltungsprozess nach Rollen, Druin's Modell, eigene Darstellung nach (Markopoulos et al., 2008, S. 46)	73
Abbildung 6: Einsatz von Prototyping, eigene Darstellung in gekürzter Form nach (Liebal & Exner, 2011, S. 191).....	78
Abbildung 7: Kapitelstruktur Praxisteil	95
Abbildung 8: Phasen der qualitativen Feldforschung	96
Abbildung 9: Die zusammenfassende Inhaltsanalyse	102
Abbildung 10: Phasen des Prozesses zur Entwicklung des Frameworks	106
Abbildung 11: Schematische Darstellung Dokumentation des Programms	114
Abbildung 12: Farbkodierung Schwierigkeitsgrade	120
Abbildung 13: Aufgabenübersicht Inhaltsdimension B	122
Abbildung 14: Aufgabenübersicht Inhaltsdimension A	123
Abbildung 15: Aufgabenübersicht Inhaltsdimension C	124
Abbildung 16: Aufgabenübersicht Inhaltsdimension D	124
Abbildung 17: Aufgabenübersicht Inhaltsdimension E	125
Abbildung 18: Tätigkeitslandkarte für das Framework	126
Abbildung 19: Testsetting Abschluss-Assessment.....	130
Abbildung 20: Vorgehen zur Definition des Kategoriensystems	133
Abbildung 21: Framework - Überblick.....	148
Abbildung 22: Bee-Bot von vorne und hinten, Quelle: TTS Group, http://www.tts-group.co.uk/bee-bot-rechargeable-floor-robot/1001794.html , 20.07.2016	174
Abbildung 23: Einfacher Zustandsautomat zum Bee-Bot, übernommen aus (Stöckelmayr, Tesar & Hofmann, 2011, S. 186).....	174
Abbildung 24: Testsetting Hardware Usability Test.....	175
Abbildung 25: Dokumentationsschema zur Transkription	177
Abbildung 26: Vorgehen zur Definition des Kategoriensystems	177
Abbildung 27: Testsetting Software Usability Test	187
Abbildung 28: Dokumentationsschema zur Transkription	191
Abbildung 29: Transkription Software Usability Test.....	191
Abbildung 30: Iteratives Closed Card Sorting Verfahren zur Hauptkategorienbildung, vorletzte Iteration mit 12 Hauptkategorien	192
Abbildung 31: Ablauf Child Centered Design	212
Abbildung 32: Anordnung Bedeutungskategorien.....	219
Abbildung 33: Zuordnung der Symbole zu den Bedeutungen.....	220
Abbildung 34: Erfassungsbogen für die Zuordnung der Symbole zu Bedeutungen durch Kind.....	220
Abbildung 35: Paper-Prototyp 1	228
Abbildung 36: Paper-Prototyp 2.....	228
Abbildung 37: Paper-Prototyp 3.....	229
Abbildung 38: Testsetting Usability Evaluierung Prototyp 1	234
Abbildung 39: Zuordnung Evaluationsfragen zu Designphasen, frei übersetzt und in eigener Darstellung nach (Markopoulos et al., 2008, S. 72).....	A-4
Abbildung 40: Perspektiven in Software Focus on Bee-Bot 2	A-48

Tabellenverzeichnis

Tabelle 1: Überblick Gestaltungsempfehlungen nach (Liebal & Exner, 2011, S. 125 - 136), eigene Darstellung	67
Tabelle 2: Zuordnung Lernergebnisse zu Kompetenzen.....	111
Tabelle 3: Leitfaden zur Kodierung Kindergarten D und E	117
Tabelle 4: Anzahl der Aufgaben pro Inhaltsdimension und Schwierigkeitsgrad	121
Tabelle 5: Anzahl der Aufgaben nach ihrer Art und Schwierigkeitsgrad.....	125
Tabelle 6: Zuordnung von Aufgaben zu Lernergebnissen.....	129
Tabelle 7: Verteilung Alter und Geschlecht der Probandinnen und Probanden.....	130
Tabelle 8: Zuordnung Fragestellung Abschluss-Assessment (AS) zu Lernergebnissen (LE)	132
Tabelle 9: Kategoriensystem Abschluss-Assessment, finale Version	136
Tabelle 10: Aktivitätsschablonen nach Leitthema und Lernergebnissen	163
Tabelle 11: Aktivitätsschablonen nach Lernergebnissen und Inhaltsdimensionen	165
Tabelle 12: Aktivitätsschablonen nach Tätigkeiten und Lernergebnissen	166
Tabelle 13: Verteilung Alter und Geschlecht der Probandinnen und Probanden.....	176
Tabelle 14: Kategoriensystem Hardware Usability Test, finale Version	179
Tabelle 15: Verteilung Alter und Geschlecht der Probandinnen und Probanden.....	187
Tabelle 16: Kategoriensystem Software Usability Test, finale Version.....	195
Tabelle 17: Ausgewählte Symbole für Card Sorting.....	218
Tabelle 18: Konsolidierte Ergebnisse Card Sorting und Bewertung gesamt	223
Tabelle 19: Finale Symbole nach Card Sorting und Bewertung	226
Tabelle 20: Übersicht Prototyp 1	232
Tabelle 21: Kategoriensystem zur standardisierten Beobachtung der Usability-Evaluation des Prototyps 1	236
Tabelle 22: Übersicht Prototyp 2	242
Tabelle 23: Kategoriensystem zur standardisierten Beobachtung der Usability-Evaluation des Prototyps 2	245
Tabelle 24: Lernergebnisse Child-Centered-Design	249
Tabelle 25: Fragen zur Evaluations- und Testgestaltung, frei übersetzt nach (Markopoulos et al., 2008, S. 14 f.).....	A-2
Tabelle 26: Aggregierte Handlungsempfehlungen zur Gestaltung von Usability-Tests nach (Hanna, Risdien & Alexander, 1997, S. 11 - 14)	A-3
Tabelle 27: Kodierung Tasten am Bee-Bot.....	A-42
Tabelle 28: Symbole der Kinder aus Gruppe 1	A-53
Tabelle 29: Symbole der Kinder aus Gruppe 2	A-55
Tabelle 30: Symbole der Kinder aus Gruppe 3	A-57
Tabelle 31: Eigene Symbole.....	A-59
Tabelle 32: Konsolidierte Ergebnisse Card Sorting und Bewertung nach 5 Kindern	A-61

Anhang

A1 Guidelines

A1.1 Fragen zur Evaluations- und Testgestaltung nach (Markopoulos et al., 2008)

Entwicklungsmerkmale	Fragen, die es zu beachten gilt
Physische Entwicklung	<ul style="list-style-type: none"> • Ist es den Testteilnehmerinnen und -teilnehmern physisch und ergonomisch möglich die Eingabegeräte oder jegliches andere Testequipment zu bedienen? • Sind die Eingabegeräte für die spezifizierte Altersgruppe geeignet? • Ist das Innendesign des Testraums sowohl für kleine als auch große Testteilnehmerinnen und -teilnehmer geeignet?
Sozioemotionale Entwicklung	<ul style="list-style-type: none"> • Ist die Testsituation für ein Kind sozial verständlich und sicher? • Kann sich ein Kind alleine, ohne einen Peer oder einen bekannten Erwachsenen, auf diese Situation einlassen? • Ist die Anzahl an anwesenden Erwachsenen und Kindern ausgewogen? • Wie sehr sind die Testteilnehmerinnen und -teilnehmer auf die Hilfe von Erwachsenen angewiesen? • Möchten Kinder mit ihrem Verhalten Erwachsenen gefallen oder sie langweilen? • Welches Verhalten der Testleitung kann die Offenheit der Testsituation steigern oder vermindern? • Kann ein Kind mit einem anderen kooperieren, wenn es die Testsituation erfordert? • Kann die Testsituation Bedrängnis oder Gefühle des Versagens bei Kindern hervorrufen und kann das die Situation und die Ergebnisse beeinflussen? • Kann die Testsituation Kindern Angst machen? Wie kann das verhindert werden? • Kann ein Kind in der Testsituation gefördert werden? Beeinflusst das die subjektive Beurteilung der Produkteigenschaften durch das Kind? • Versteht ein Kind, dass das Produkt evaluiert wird und nicht das Kind? Wie fühlt sich das Kind dabei, Teil eines Tests zu sein? • Empfindet ein Kind ein positives Gefühl in Bezug auf den Test, wenn dieser vorbei ist?
Kognitive Entwicklung	<ul style="list-style-type: none"> • Versteht ein Kind, was von ihm erwartet wird? • Versteht ein Kind, was es bedeutet „eine Aufgabe auszuführen“? • Sind die Aufgabenstellungen verständlich und einprägsam? • Kann ein Kind das Produkt alleine benutzen? • Versteht ein Kind Ursache-Wirkungs-Beziehungen, die notwendig sind für die Nutzung des Produkts? • Sind die Problemlösungskompetenzen eines Kindes ausreichend ausgeprägt um die Aufgabenstellung zu lösen? • Weist die Testsituation Eigenschaften auf, die Kinder vom Denken und ihrer Aufmerksamkeit ablenken können? • Ist die kognitive Belastung zu hoch für Kinder? Tragen die gewählten Testmethoden zusätzlich dazu bei? • Kann sich ein Kind an bisherige Aktivitäten und Gedanken erinnern? Können retrospektive Methoden verwendet werden, die darauf aufbauen? • Kann ein Kind lesen? Wenn nicht, wie werden die Aufgabenstellungen dem Kind präsentiert und wie soll eine erwachsene Person das Kind unterstützen,

	<p>wenn dieses die Aufgabenstellung vergessen hat?</p> <ul style="list-style-type: none"> • Kann schriftliche Sprache als Teil des Tests verwendet werden? • Versteht ein Kind die Sprache und die Konzepte, die die Testleitung verwendet? • Kann ein Kind schreiben? Kann ein Kind schriftlich antworten? • Wie gut kann ein Kind Gedanken verbalisieren? • Kann ein Kind auf Fragen während eines Interviews antworten? Beinhalten die Antworten des Kindes ausreichend Information für die Evaluation?
--	---

Tabelle 25: Fragen zur Evaluations- und Testgestaltung, frei übersetzt nach (Markopoulos et al., 2008, S. 14 f.)

A1.2 Handlungsempfehlungen zur Gestaltung von Usability-Tests nach (Hanna, Ridsen & Alexander, 1997)

Testphase	Handlungsempfehlungen
Testgestaltung und Planung	<ul style="list-style-type: none"> • Der Testraum ist kinderfreundlich zu gestalten und sollte nicht zu überladen sein, um Ablenkung zu vermeiden. • Junge Kinder haben Probleme beim Wechsel der Eingabegeräte, wenn sie zu Hause andere verwenden als im Testszenario. Darüber hinaus sollten die Eingabegeräte auf die robusteste Einstellung konfiguriert werden. • Testequipment soll so effektiv wie möglich und so unaufdringlich wie möglich eingesetzt werden. • Für ältere Kinder soll eine Testzeit von einer Stunde und für junge Kinder eine Testzeit von 30 Minuten eingeplant werden. Darüber hinaus ist zu berücksichtigen, dass Kinder zusätzliche Zeit zum Spielen und Entdecken benötigen. • Wenn eine Serie von Aufgaben geplant wird, dann sollte die Reihenfolge variieren, damit nicht immer die gleichen Aufgaben zum Schluss kommen, wenn die Aufmerksamkeit der Kinder bereits nachlässt. (Ausgenommen man möchte genau eine bestimmte Reihenfolge testen.) • Die Kinder, die an den Tests teilnehmen, sollten zumindest ein wenig Erfahrung in der Bedienung eines Computers haben, sonst muss während des Tests die grundlegende Bedienung erklärt werden. (Ausnahme, wenn man genau dies testen möchte.) • Die Kinder, die an den Tests teilnehmen, sollten auch nicht über zu viel Erfahrung in der Bedienung eines Computers verfügen, da sonst ihr Erfahrungswissen über mögliche Usability-Probleme wegtäuschen könnte. • Die Kinder sollten nicht aus dem näheren Umfeld der Testleitung stammen, da diese wahrscheinlich über mehr Kenntnisse im Kontext der zu testenden Produkte verfügen, als der Durchschnitt aller anderen Kinder.
Einführung in den Test	<ul style="list-style-type: none"> • Zu Beginn ist eine Beziehung zu dem Kind aufzubauen. Ein gegenseitiges Kennenlernen soll im Rahmen von kurzem Small-Talk geschehen. • Allfällige Geheimhaltungen bei neuen Produktentwicklungen können durch das Entwickeln eines „Geheimnisses“ mit dem Kind vor Veröffentlichungen bewahrt werden. Eltern sollten eine Geheimhaltungserklärung unterzeichnen. • Folge einem vorbereiteten Skript, um Kinder in das Testszenario einzuführen. Nur so kann sichergestellt werden, dass alle Testteilnehmerinnen und -teilnehmer zu Beginn die gleichen Informationen erhalten. • Ältere Kinder können durch die Darstellung und Erläuterung der Bedeutung ihrer Rolle zur Teilnahme motiviert werden. • Es ist darauf zu achten, dass Kinder keine überzogenen Erwartungshaltungen an die Testinhalte haben – wenn ein Prototyp getestet werden soll, dann sollte das den Kindern auch erläutert werden. Nur so kann eine allfällige Enttäuschung der

	<p>Kinder über das Nichtvorhandensein eines fertigen Produkts vermieden werden.</p> <ul style="list-style-type: none"> • Der Testraum und das Testequipment sollten sowohl den Kindern als auch den Erwachsenen erklärt und auf alle Kameras und Beobachterinnen und Beobachter hingewiesen werden. • Junge Kinder sollten nicht allein im Testraum sein, die Testleitung oder eine anleitende Person, die den Test durchführt, sollte sich mit dem Kind im Raum befinden. • Bei jungen Kindern, die schüchtern sind oder sich in Gegenwart von Fremden unwohl fühlen, muss es möglich sein, dass auch die Eltern oder eine Vertrauensperson beim Test anwesend sein können. • Wenn Geschwister die Kinder zum Test begleiten, so sollten diese in einem separaten Raum auf die Testteilnehmerin, den Testteilnehmer warten, denn gerade ältere Geschwister möchten ihren jüngeren Geschwistern gerne Anweisungen und Tipps geben.
Während des Tests	<ul style="list-style-type: none"> • Jungen Kindern muss ein wenig Zeit zur Gewöhnung an den Computer oder Testgegenstand eingeräumt werden. • Ältere Kinder können Aufgaben lösen, die durch freies Probieren und Entdecken gelöst werden. • Kinder fragen oftmals um Hilfestellung. Diese Fragen sind mit Gegenfragen, die die Problemlösungskompetenzen des Kinds anregen, zu beantworten, um nicht die Lösung von Aufgaben vorzugeben. • Kinder sollen nicht gefragt werden, ob sie die Aufgabe lösen oder das Spiel spielen wollen - sie könnten mit „Nein“ antworten. Besser sind Phrasen wie: „Lass uns jetzt...“, „Es ist Zeit um...“, „Nun möchte ich, dass du...“ • Wenn die Aufmerksamkeit der Kinder nachlässt, soll diese mit motivierenden Aufforderungen wiedererlangt werden. • Wenn Kinder (noch) nicht lesen können, sind die jeweiligen Texte und Zahlen durch die Testleitung vorzulesen. • Kinder sollen durch generische positive Rückmeldungen motiviert werden.
Abschluss des Tests	<ul style="list-style-type: none"> • Die Begeisterung und Motivation der Kinder zur Benutzung des Programms oder Produkts ist durch entsprechende Methoden am Ende zu messen. • Ältere Kinder können zum Abschluss umfangreichere Befragungen oder Fragebögen beantworten. • Zum Abschluss soll die Bedeutung der Mitwirkung der Kinder positiv hervorgehoben und betont werden. • Die Teilnahme am Test soll durch ein kleines Geschenk belohnt werden. Idealerweise kann das Kind aus einer kleinen Auswahl wählen.

Tabelle 26: Aggregierte Handlungsempfehlungen zur Gestaltung von Usability-Tests nach (Hanna, Ridsen & Alexander, 1997, S. 11 - 14)

A1.3 Zuordnung Evaluationsfragen zu Designphasen nach (Markopoulos et al., 2008)

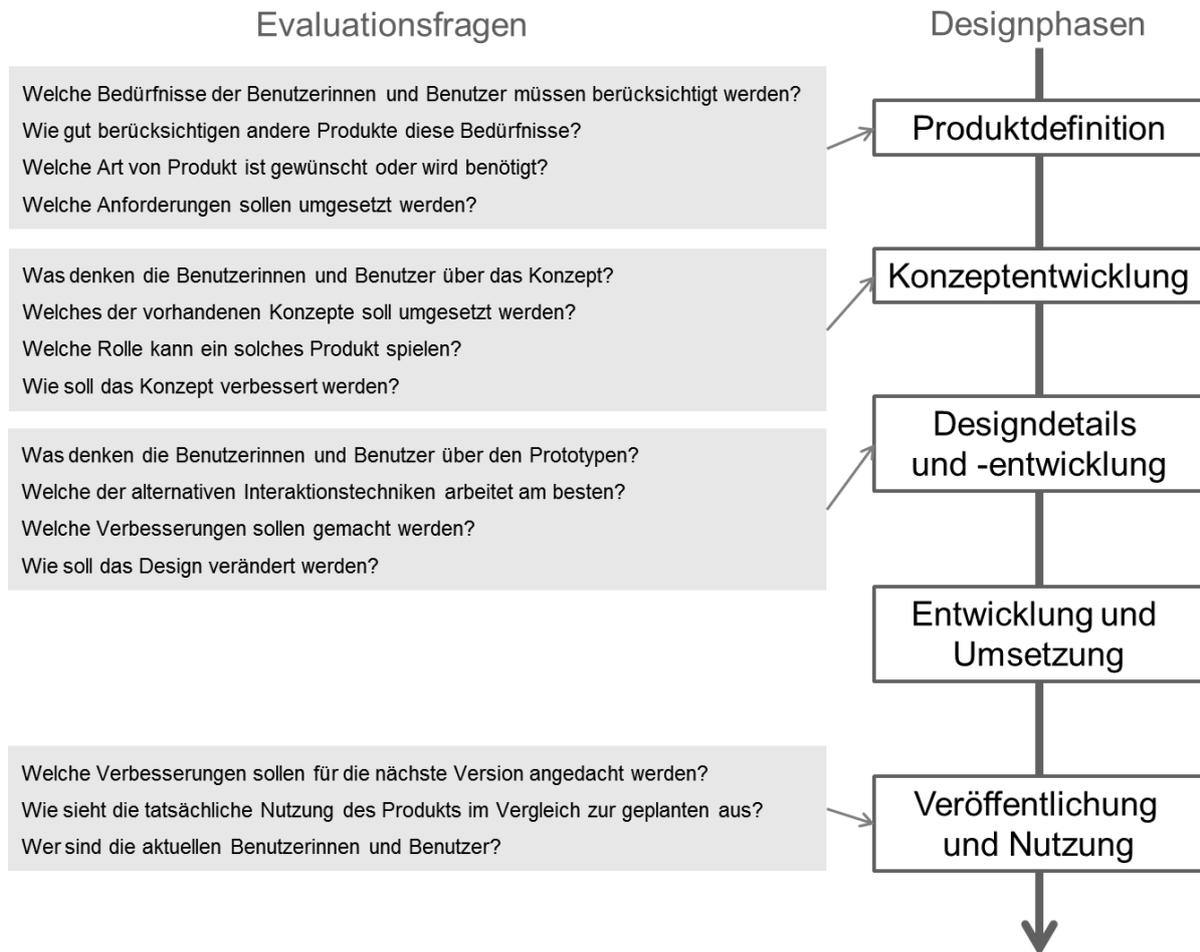


Abbildung 39: Zuordnung Evaluationsfragen zu Designphasen, frei übersetzt und in eigener Darstellung nach (Markopoulos et al., 2008, S. 72)

A2 Konkrete durchgeführte Aufgaben der Studie

A2.1 Aufgaben Inhaltsdimension A

AufgabenID:	11	Schwierigkeitsgrad:	3	Dauer:	2 Minuten
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Ein Kind bekommt in der Gruppe Aufgabe mit Bee-Bot zu anderem Kind im Sitzkreis zu fahren</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot. Alle sitzen im Sitzkreis. Kinder bekommen reihum die Aufgabe mit dem Bee-Bot zu anderem Kind im Sitzkreis zu fahren.				
Positive Notiz:	Kinder entwickeln eigene Lösungsstrategien für Entfernungsabschätzung, Zählen im Zahlenraum größer 10.				
Negative Notiz:	Eingabe einer willkürlichen Anzahl von Vorwärtsbefehlen ohne Planung, abweichen von vorheriger Planung der Schritte bei der Eingabe.				
Allgemeine Erkenntnis:	Hier erwähnen / danach fragen, dass / ob die Schritte, die der Bee-Bot fährt immer gleich groß sind. Kinder planen am Boden die notwendigen Schritte bis zum Ziel. Manche Kinder behelfen sich hier mit im Raum vorhandenen Anhaltspunkten, wie z.B. gemasertem Parkettboden zum Abschätzen, wie viele Schritte notwendig sind. Hinweis -> ein Schritt des Bee-Bots entspricht ca. einer Handlänge. Hinweis: Wenn alle Kinder drankommen sollen, empfiehlt sich davor eine zwischenzeitliche Pause für WC-Besuch und trinken. Außerdem benötigen die Kinder Unterstützung, wer nach der Reihe drankommen soll.				

AufgabenID:	12	Schwierigkeitsgrad:	2	Dauer:	10 Minuten
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Vertraut machen mit Bee-Bot durch Auspacken, freies Üben und Erkunden der Funktionalität des Bee-Bots - einzeln ohne vorgegebene "Spielfläche"</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: entweder für jedes teilnehmende Kind oder immer für zwei Kinder ein aufgeladener Bee-Bot - keinesfalls nur ein Zweier-Team bilden - Kinder dieses Teams fühlen sich benachteiligt. Die Kinder bekommen Bee-Bots in Schachteln ausgeteilt und werden zum Ausprobieren und Erkunden ermuntert.				
Positive Notiz:	Ideenreichtum, versinken im Tun, Hardware-Robustheit.				
Negative Notiz:	Unsanfter Umgang mit Bee-Bot, schnelles Benachteiligungsgefühl.				
Allgemeine Erkenntnis:	Einige Kinder benötigen Unterstützung beim Einschalten / Bedienen des Bee-Bots. Genügend großen Raum vorsehen.				

AufgabenID:	17	Schwierigkeitsgrad:	2	Dauer:	15 Minuten
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Freies Üben und Erkunden der Funktionalität des Bee-Bots auf vorgegebener „Spielfläche“: Matten (pro Kind ein Bee-Bot, Kinder aufgeteilt auf drei Matten)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: drei vorgefertigte Matten mit jeweils ausreichend Platz rund um diese, eine Betreuungsperson pro Matte (bei weniger Betreuungspersonal Mattenanzahl reduzieren), entweder für jedes Kind oder für je zwei Kinder einen aufgeladenen Bee-Bot. Die Kinder werden auf die Matten aufgeteilt und dürfen dort mit den Bee-Bots die Wege fahren, die sie möchten und bekommen dabei, falls erforderlich, Unterstützung. Nach einer vorgegebenen Zeit wechseln die Kinder zur nächsten Matte.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Absichtliche Bee-Bot-Kollisionen, kein versinken im Tun durch Angebot unterschiedlicher Matten auf den verschiedenen Stationen.				
Allgemeine Erkenntnis:	Sollte ein Kind bei der Einführung zum Bee-Bot gefehlt haben – verstärkte Unterstützung sicherstellen. Genügend Platz am Boden rund um jede Matte erforderlich.				

AufgabenID:	119	Schwierigkeitsgrad:	1	Dauer:	7 Minuten
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Erforschen				
Aufgabe:	<i>Kennenlernen der Symbolmatte in der Gruppe - jedes Kind wählt sein Lieblingssymbol aus.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Symbolmatte. Alle sitzen im Sitzkreis. Bevor die Bee-Bots zum Fahren auf der Matte programmiert werden, findet bei dieser Aufgabe ein „Kennenlernen“ der Matte statt, indem jedes Kind sagt, welches Symbol ihm am besten gefällt und was es darstellt. Daran anknüpfend kann erwähnt werden, dass „der Bienenroboter jetzt auch die Matte kennenlernen möchte“ was als				

	Überleitung zum Fahren auf der Matte fungiert.
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	148	Schwierigkeitsgrad:	4	Dauer:	10 Minuten
Aktivitätscluster:	A1) Exploration Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Präsentieren				
Aufgabe:	<i>Kinder erklären und zeigen gemeinsam (moderiert und unterstützt von TR) KindergartenPäd. die Funktionalität des Bee-Bots durch Lösen von Aufgaben (von Anfangs- zu Endsymbol fahren) auf der Schatzinselmatte (4mal4 Felder), sitzend im Sitzkreis.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Schatzinselmatte. Alle sitzen im Sitzkreis, Kindergartenpädagogin / -pädagoge sitzt mit im Kreis. Kinder erklären und zeigen gemeinsam (moderiert und unterstützt von TR) KindergartenPäd. die Funktionalität des Bee-Bots durch Lösen von Aufgaben (von Anfangs- zu Endsymbol fahren) auf der Schatzinselmatte.				
Positive Notiz:	Kinder genießen sichtlich Rollentausch und beteiligen sich aktiv, zeigen unterschiedliche Lösungsstrategien.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	10	Schwierigkeitsgrad:	3	Dauer:	10 Minuten
Aktivitätscluster:	A2) Vorstellung Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Zeigen, erklären und vorführen des Bee-Bot durch TR</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot. Alle sitzen im Sitzkreis. TR zeigt und erklärt den Bee-Bot und führt ihn dann vor. Im Zuge der Erklärungen der Befehlstasten sollen auch Hilfestellungen für Richtungsunterscheidungen gegeben werden – z.B. vorwärts ist dort, wo der Bee-Bot hinschaut, Unterscheidung links und rechts erarbeiten.				
Positive Notiz:	Interesse, Konzentration				
Negative Notiz:	Problem bei Links- / Rechtsunterscheidung.				
Allgemeine Erkenntnis:	Im Zuge der Erklärung des Bee-Bots kommen auch die Richtungen Links und Rechts vor. Auf Nachfrage, ob die Kinder diese Richtungen schon kennen antworten die meisten Kinder mit Ja. Sie wissen diese Richtungen von oben und unten zu unterscheiden, haben allerdings oftmals noch Probleme bei der Unterscheidung von links und rechts. Kinder auf Merkmale wie Armbänder, etc. als Anker hinweisen um ihnen die Unterscheidung zu erleichtern. Hilfestellung: Befehl wurde eingespeichert, wenn die Augen des Bee-Bots aufleuchten. Unterscheidung Clear und Pause, z.B. Clear beginnt mit einem halben Kreis (C).				

AufgabenID:	15	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A2) Vorstellung Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Erstes Fahren des Bee-Bots auf Matte – zeigen, dass Schritte immer gleich groß wie ein Kästchen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot und eine vorgefertigte Matte (z.B. Symbolmatte) mit Kästchen in Schrittweite des Bee-Bots. Alle sitzen im Sitzkreis. TR zeigt von mehreren unterschiedlichen Startpositionen aus vor, dass ein Vorwärtsschritt immer das Wechseln des Bee-Bots aufs nächste Feld bewirkt – zwei Vorwärtsschritte eine Fahrt zwei Felder weiter bewirken, usw.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Die Symbolmatte empfiehlt sich für das Vorzeigen dieses Sachverhalts.				

AufgabenID:	116	Schwierigkeitsgrad:	4	Dauer:	10 Minuten
Aktivitätscluster:	A2) Vorstellung Programmierumgebung			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Zeigen, erklären und vorführen des Bee-Bots durch TR inkl. Anknüpfen an Erfahrungen mit Befehlskarten</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: ein aufgeladener Bee-Bot, Befehlskarten. Alle sitzen im Sitzkreis. Zeigen, erklären und vorführen des Bee-Bots durch TR inkl. Anknüpfen an Erfahrungen mit Befehlskarten.				
Positive Notiz:	Anknüpfen an Befehlskartenerfahrungen der Kinder.				
Negative Notiz:	Problem bei Links- / Rechtsunterscheidung.				

Allgemeine Erkenntnis:	Im Zuge der Erklärung des Bee-Bots kommen auch die Richtungen Links und Rechts vor. Auf Nachfrage, ob die Kinder diese Richtungen schon kennen antworten die meisten Kinder mit Ja. Sie wissen diese Richtungen von oben und unten zu unterscheiden, haben allerdings oftmals noch Probleme bei der Unterscheidung von links und rechts. Kinder auf Merkmale wie Armbänder, etc. als Anker hinweisen um ihnen die Unterscheidung zu erleichtern. Hilfestellung: Befehl wurde eingespeichert, wenn die Augen des Bee-Bots aufleuchten. Unterscheidung Clear und Pause, z.B. Clear beginnt mit einem halben Kreis (C).
-------------------------------	--

AufgabenID:	16	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	ganze Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Rechtsfahren auf Straßenmatte in der Gruppe</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot und die vorgefertigte Straßenmatte. Alle sitzen im Sitzkreis. TR stellt den Bee-Bot vor die Kreuzung, der Bee-Bot soll nach rechts abbiegen (ein Schritt vorwärts, Vierteldrehung nach rechts, ein Schritt vorwärts). Diese Aufgabe wird mit allen Kindern gemeinsam gelöst, TR gibt die Befehle ein, die die Kinder vorschlagen.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Fehlende Disziplin, Ungeduld.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	120	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Fledermaus, Blick nach rechts, zu Endsymbol – Hut, Blick nach rechts) Kind im Sitzkreis auf Symbolmatte: V</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Perspektivenprobleme				
Allgemeine Erkenntnis:	Matte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder: sich direkt hinter den Bee-Bot auf die Matte (oder außen nahe zur Matte hinter den Bee-Bot) setzen, damit die Blickrichtungen des einspeichernden Kindes und des Bee-Bots übereinstimmen und so die Wegplanung leichter fällt.				

AufgabenID:	121	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Hut, Blick nach rechts, zu Endsymbol – Katze, Blick nach rechts) Kind im Sitzkreis auf Symbolmatte: VV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Hohe Aufmerksamkeit aller Kinder (auch der zusehenden), Kind löst Aufgabe ohne Hilfestellung, anderes Kind korrigiert selbstständig falsche Wegespeicherung ohne Hinweis.				
Negative Notiz:	Perspektivenprobleme				
Allgemeine Erkenntnis:	Matte am besten mit Klebeband am Boden befestigen. Hilfestellung für die Kinder: sich direkt hinter den Bee-Bot auf die Matte (oder außen nahe zur Matte hinter den Bee-Bot) setzen, damit die Blickrichtung des einspeichernden Kindes und des Bee-Bots übereinstimmen und so die Wegplanung leichter fällt. Durch das Stellen der Aufgaben durch TR ist ein gezieltes Steigern des Schwierigkeitsgrades möglich (Aufgabe ID120, ID121, usw.) – durch die Beobachtung der anderen Kinder in der Gruppe beim Lösen dieser Aufgabe lernen alle Kinder langsam das Aneinanderreihen der Befehle. Durchgehend bei allen Aufgaben dieses Typs (von TR gestellte Einzelaufgabe auf Symbolmatte, Kind im Sitzkreis) sehr hohe Aufmerksamkeit bei allen Kindern - auch bei den zusehenden.				

AufgabenID:	122	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Katze, Blick nach rechts, zu Endsymbol – Nonne, Blick nach unten) Kind im Sitzkreis auf Symbolmatte: VRVV</i>				

Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Andere Kinder geben sinnvolle Hilfestellungen.
Negative Notiz:	Kind hört nicht auf Hilfestellungen der anderen Kinder, Funktionalität des Bee-Bots noch nicht verstanden, zählt beim Abzählen der Vorwärtsschritte Startfeld mit, Kind ist Erhalt der gespeicherten Befehle trotz zwischenzeitlicher Fahrt nicht klar.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	124	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Hai, Blick nach unten, zu Endsymbol – Fuchs, Blick nach oben) Kind im Sitzkreis auf Symbolmatte: RVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind nimmt Perspektive des Bee-Bots ein, hohe Aufmerksamkeit aller Kinder.				
Negative Notiz:	Links-Rechts-Probleme bezüglich Bezeichnungen, zählt beim Abzählen der Vorwärtsschritte Startfeld mit, Kind durch Methode „Trockenweiterstellen“ hinsichtlich des Speicherinhalts irritiert, Funktionalität von Go scheint nicht klar.				
Allgemeine Erkenntnis:	Bei dieser Aufgabe ansprechen, dass es mehrere Wege von A nach B gibt und Kind dazu anregen, vor dem Einspeichern der Befehle den geplanten Weg anzugeben.				

AufgabenID:	125	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Fuchs, Blick nach oben, zu Endsymbol – Fledermaus, Blick nach links) Kind im Sitzkreis auf Symbolmatte: VVVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit kleiner Hilfestellung ohne „Trockenweiterstellen“.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	126	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Fledermaus, Blick nach links, zu Endsymbol – Sonne, Blick nach links) Kind im Sitzkreis auf Symbolmatte: RürRürRürRürRür</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit Hilfestellung ohne „Trockenweiterstellen“.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	127	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Sonne, Blick nach links, zu Endsymbol – Fledermaus, Blick nach links) Kind im Sitzkreis auf Symbolmatte: VVVVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbol des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und				

	US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Hilfestellung von anderen Kindern.
Negative Notiz:	Befehlseingabe des Kindes vor Nennen des Endsymbols.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	128	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von TR gestellte Einzelaufgabe (von Anfangssymbol – Fledermaus, Blick nach links, zu Endsymbold – Bus, Blick nach rechts) Kind im Sitzkreis auf Symbolmatte: LVVVVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbold des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Für geplanten Weg des Kindes andere Drehungen notwendig als 90 Grad, Kind zählt über alle Felder hinweg die Gesamtfelderdistanz, die Bee-Bot bewältigen muss – ungeachtet der Richtungen / Drehungen, viel Hilfestellung.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	129	Schwierigkeitsgrad:	4	Dauer:	6 Minuten
Aktivitätscluster:	A3) Gestelltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Von allen gemeinsam gestellte Einzelaufgabe (von Anfangssymbol – Bus, Blick nach rechts, über Zwischensymbol – Fuss, Blick nach oben, zu Endsymbold – Spinnennetz, Blick nach links) Kind im Sitzkreis auf Symbolmatte: VVVVVLV(hier Zwischensymbol)VVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, vorgefertigte Symbolmatte. Alle sitzen im Sitzkreis. TR gibt das Anfangs- und Endsymbold des Weges vor, den das Kind mit dem Bodenroboter auf der Arbeitsmatte befahren soll. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit Hilfestellung mit „Trockenweiterstellen“.				
Negative Notiz:	Kind zählt über alle Felder hinweg die Gesamtfelderdistanz, die Bee-Bot bewältigen muss – ungeachtet der Richtungen / Drehungen, zählt Startfeld mit, viel Hilfestellung.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	27	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Fahnen zu Endsymbold – Menschen) Kind im Sitzkreis auf selbstgezeichneter Matte: VVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbold des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Planung der Programmierschritte im Gedächtnis.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	28	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Giraffe zu Endsymbold – Fische) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVRVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbold des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				

Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Keine Differenzierung zwischen Richtungen bei Planung der Programmierschritte, Ausgangsfeld wird als notwendiger Schritt mitgezählt, Erhalt des Speicherinhalts nach Drücken GO-Taste wurde vergessen, zu schwierige eigene Aufgabenstellung.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	29	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Menschen zu Endsymbol – Fahnen) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVLVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Viele Hilfestellungen der Kinder, Perspektivenwechsel zwischen Bee-Bot und Kind geglückt.				
Negative Notiz:	Zu schwierige eigene Aufgabenstellung – nur mit viel Hilfestellung gelöst, Teiletappenplanung, Gesamtweg nur mit Hilfestellung.				
Allgemeine Erkenntnis:	Höherer Schwierigkeitsgrad, wenn die Strecke in einem durchgefahren werden soll und nicht Zwischenstopps an neuralgischen Punkten mit zwischenzeitlichem Löschen des Speichers eingelegt werden können.				

AufgabenID:	31	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Fische zu Endsymbol – Restaurant / Getränke und Kuchen, außen herum, nicht direkter Weg) Kind im Sitzkreis auf selbstgezeichneter Matte: RVVVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst durch eigene Korrekturen die Aufgabe.				
Negative Notiz:	Kind fährt Umweg, zu schwierige eigene Aufgabenstellung.				
Allgemeine Erkenntnis:	Wird die Zusatzforderung des kürzesten Weges gestellt, erhöht sich der Schwierigkeitsgrad, wobei der lange Weg außen herum in diesem Fall schwieriger war. TR wollte durch Eingreifen mit Vorschlag kürzesten Weg die schwierige Aufgabenstellung des Kindes selbst mildern – hat nicht funktioniert, da Kind auf langen Weg bestand.				

AufgabenID:	32	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Fahnen zu Endsymbol – Fahnen) Kind im Sitzkreis auf selbstgezeichneter Matte: Kind zeigt Serpentinenkurs auf der Matte an, fährt dann allerdings eine Außenrunde (VVVRVVVRVVVRVVV)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind schafft Programmierung des Weges ohne Hilfestellung.				
Negative Notiz:	Kind fährt anderen Weg als den zuvor geplanten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	33	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Restaurant zu Endsymbol – Fahnen, nicht direkter Weg, Umweg über Giraffe) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVLVVLVVV</i>				

Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Problem bei Links- / Rechtsunterscheidung, Verweigerung weiterzulösen als Fehler gemacht wird.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	35	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Haus zu Endsymbol – Menschen) Kind im Sitzkreis auf selbstgezeichneter Matte: VRVLVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind bleibt trotz Problemen motiviert und probiert weiter die Aufgabe zu lösen.				
Negative Notiz:	Erhalt des Speicherinhalts nach zwischenzeitlichem Drücken GO-Taste wurde vergessen, zu schwierige eigene Aufgabenstellung, wählt komplizierten Weg von A nach B, keine Differenzierung zwischen Richtungen bei Planung der Programmierschritte				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	36	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Haus zu Endsymbol – Frau mit vier Kindern) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVRVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Trotz voriger Verweigerung motiviert sich Kind neue Aufgabe zu lösen, löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung.				
Negative Notiz:	Vergisst vor Beginn der Programmeinspeicherung den Speicher zu löschen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	37	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – Frau mit vier Kindern, Blick nach unten, zu Endsymbol – Feld mit Kuchen und Getränken) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Trotz voriger Frustration motiviert sich Kind neue Aufgabe zu lösen, löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung, ein anderes Kind löst die Aufgabe ohne Methode „Trockenweiterstellen“ mit Hilfestellung.				
Negative Notiz:	Keine Differenzierung zwischen Richtungen bei Planung der Programmierschritte, schieben des Bee-Bots im ab- oder auch eingeschalteten Zustand um ihn zu positionieren.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	39	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe

Aufgabenart:	Programmieren
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – A5 mit Blick nach unten zu Endsymbol – D4) Kind im Sitzkreis auf selbstgezeichneter Matte: VLVVV</i>
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung.
Negative Notiz:	Nicht notwendige Vereinfachung der Aufgabe durch Methode „Trockenweiterstellen“ (Kind löste frühere Aufgabe erfolgreich durch direkte Eingabe ohne Weiterstellen), Hinweis von TR wäre gut gewesen.
Allgemeine Erkenntnis:	Selbstgezeichnete Matten tauschen, sodass nacheinander alle selbstgezeichneten Matten vom Bee-Bot „befahren“ werden.

AufgabenID:	40	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – C1 mit Blick nach oben, zu Endsymbol – C5, fährt entlang der Straße) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVLVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung.				
Negative Notiz:	Nicht notwendige Vereinfachung der Aufgabe durch durchgängige Methode „Trockenweiterstellen“, Hinweis von TR wäre gut gewesen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	41	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – A6 mit Blick nach unten zu Endsymbol – D1, eigenwilliger Weg außerhalb der Straße) Kind im Sitzkreis auf selbstgezeichneter Matte: VLVVVRVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung.				
Negative Notiz:	Vergisst vor Beginn der Programmeinspeicherung den Speicher zu löschen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	42	Schwierigkeitsgrad:	4	Dauer:	5 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – A1 mit Blick nach oben zu Endsymbol – C1 mit Blick nach unten, entlang der ganzen eingezeichneten Straße auf der Matte) Kind im Sitzkreis auf selbstgezeichneter Matte: VVRVVLVVRVVRVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ ohne Hilfestellung.				
Negative Notiz:	Vergisst vor Beginn der Programmeinspeicherung den Speicher zu löschen, Unsicherheiten bei Drehungen in Hinblick auf notwendigen zusätzlichen Vorwärtsschritt.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	43	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – D2 mit Blick nach links zu Endsymboll – C6 mit Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte:RVVVVLV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymboll des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ mit Hilfestellung langsamer zu arbeiten um Fehler zu vermeiden.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	44	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymboll – D3, Blick nach links, zu Endsymboll – A6, Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte:RVVVVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymboll des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ mit Hilfestellung bezüglich Startposition der Bee-Bot Fahrt nach Einspeichern.				
Negative Notiz:	Kind wusste Ausgangsblickrichtung des Bee-Bots am Startfeld nach Einspeichern nicht mehr – benötigte Hilfestellung.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	45	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymboll – D6, Blick nach unten, zu Endsymboll – D1, Blick nach unten) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymboll des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Versucht einfache Aufgabe ohne „Trockenweiterstellen“ zu lösen – löst Aufgabe mit Hilfestellung.				
Negative Notiz:	Erhalt des Speicherinhalts nach zwischenzeitlichem Drücken der GO-Taste wurde vergessen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	46	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymboll – B6, mit Blick nach unten, zu Endsymboll – B2, mit Blick nach unten) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymboll des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst im zweiten Versuch nach Hilfestellung bezüglich des einfacheren Weges die Aufgabe ohne Hilfestellung ohne Methode „Trockenweiterstellen“.				
Negative Notiz:	Kind kann Clear- und Pause-Taste nicht unterscheiden. Kind wählt schwierigere Route, die für Kind aufgrund der Drehungen sichtbar zu schwer ist. Kind zählt zu befahrende Felder ab und speichert Vorwärtsbefehle ungeachtet der tatsächlich benötigten Richtung.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

Erkenntnis:					
AufgabenID:	47	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – D6, Blick nach links, zu Endsymbol – A2) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVLVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Löst mit Methode „Trockenweiterstellen“ mit Hilfestellung sich zum Einspeichern der Befehle hinter den Bee-Bot zu setzen.				
Negative Notiz:	Verwechselt vorwärts mit GO. Andere Kinder stören und verfolgen die Aufgabenlösung des Kindes nicht.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	133	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – H5, Blick nach unten, zu Endsymbol – F1, Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVVRVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Gute Konzentrationsfähigkeit.				
Negative Notiz:	Probleme beim Planen und Eingeben der Befehle, benötigt Hilfestellung beim Zerlegen in Teilstrecken und beim Korrigieren von Fehleingaben. Kind zählt über alle Felder hinweg die Gesamtfelderdistanz, die Bee-Bot bewältigen muss – ungeachtet der Richtungen				
Allgemeine Erkenntnis:	Kinder als Hilfestellung auffordern sich beim Einspeichern der Befehle hinter die Biene zu setzen, um das Einspeichern der Befehle durch die gleiche Blickrichtung des Kindes und des Bodenroboters zu erleichtern. Kind daran erinnern, diese Sitzposition auch während des Einspeicherns laufend anzupassen wenn erforderlich.				

AufgabenID:	134	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – H5, Blick nach unten, zu Endsymbol – G1, Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Erhalt des Speicherinhalts nach zwischenzeitlichem Drücken der GO-Taste wurde vergessen -> führt zu Irritation u Kind verwechselt dann Clear- und GO-Taste.				
Allgemeine Erkenntnis:	Manche Kinder wählen sich selbst zu schwierige Aufgaben für ihren derzeitigen Übungsstand aus und benötigen sowohl sehr viel Unterstützung als auch einiges an Zeit zur Lösung der Aufgabe. Dies wirkt sich auf die Aufmerksamkeit der restlichen Kinder teilweise negativ aus. Kinder können bei Fehleingaben oder wenn der Bee-Bot von der Matte runterfährt etc. nochmals gezeigt bekommen, dass ein Drücken der Taste GO während der Fahrt des Bee-Bots ein Stoppen bewirkt (Ein erneutes GO startet mit dem Abarbeiten der Befehle wieder von vorne, nicht bei der Unterbrechung).				

AufgabenID:	135	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – D5, Blick nach unten, zu Endsymbol – C4,</i>				

	<i>Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte: VRV</i>
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Kind vergisst aufs Befehle einspeichern und schiebt anstelle dessen Bee-Bot von A nach B, benötigt viel Hilfestellungen (Wegplanung, Speicherkonzept, etc.) beim Lösen der Aufgabe mit „Trockenweiterstellen“.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	136	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – A4, Blick nach rechts, zu Endsymbol – B2, Blick nach unten) Kind im Sitzkreis auf selbstgezeichneter Matte: VRVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Kind benötigt viel Hilfestellung beim Lösen der Aufgabe mit „Trockenweiterstellen“.				
Allgemeine Erkenntnis:	Wenn alle Kinder nacheinander auf den selbstgezeichneten Matten Aufgaben lösen sollen, empfiehlt es sich nach ca. der Hälfte der Kinder eine Pause einzulegen – WC und Trinkpause. Wenn eine längere zusammenhängende Einheit geplant wird, müssen viele hinsichtlich ihrer Sozialform, Aufgabenart, etc. unterschiedliche Aufgaben in dieser Einheit kombiniert werden, da sonst die Aufmerksamkeit der Kinder nicht gehalten oder wieder erhöht werden kann.				

AufgabenID:	137	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – A2, Blick nach rechts, zu Endsymbol – G4, Blick nach oben) Kind im Sitzkreis auf selbstgezeichneter Matte: VVVVVVLVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe ohne „Trockenweiterstellen“ mit Hilfestellung.				
Negative Notiz:	Kind zählt über alle Felder hinweg die Gesamtfelderdistanz, die Bee-Bot bewältigen muss – ungeachtet der Richtungen / Drehungen, benötigt Hilfestellung beim Zerlegen des Weges in Teilstrecken.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	140	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Anfangssymbol – H2, Blick nach links, zu Endsymbol – E2, Blick nach links) Kind im Sitzkreis auf selbstgezeichneter Matte: VVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit kleiner Hilfestellung.				
Negative Notiz:	Speicher löschen durch „Clear“ wurde zuerst vergessen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	149	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen:			Sozialform:	Einzel in ganzer Gruppe

	Sequenzen		
Aufgabenart:	Programmieren		
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endsymbol der Aufgabe zuvor – D1, Blick nach oben, zu selbstgewähltem Endsymbol – A4, Blick nach oben, Kind fährt ZickZackKurs, nicht direkt) Kind im Sitzkreis auf Schatzinselmatte: VLVRVLVVRV</i>		
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Schatzinselmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.		
Positive Notiz:	Keine positiven Auffälligkeiten.		
Negative Notiz:	Viel Unterstützung beim Planen und Einspeichern des Weges erforderlich.		
Allgemeine Erkenntnis:	Keine Anmerkungen.		

AufgabenID:	150	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition von der Aufgabe zuvor – A4, Blick nach oben, zu selbstgewähltem Endsymbol – D1, Blick nach unten) Kind im Sitzkreis auf Schatzinselmatte: RürVVVVRVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Schatzinselmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Probleme beim Erkennen der Befehlstasten, zählt Startfeld mit.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	151	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition von der Aufgabe zuvor – D1, Blick nach oben, zu selbstgewähltem Endsymbol – D4, Blick nach oben) Kind im Sitzkreis auf Schatzinselmatte: VVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Schatzinselmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Speicher löschen durch „Clear“ und Befehl „GO“ zum Beginn des Abarbeitens der gespeicherten Befehle wurde zuerst vergessen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	152	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von selbstgewählter Anfangsposition – D4, Blick nach unten, zu selbstgewähltem Endsymbol – A4, Blick nach oben) Kind im Sitzkreis auf Schatzinselmatte: VVVRVVVRVVV (Dreiviertelrunde um die Matte)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Schatzinselmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind nimmt Perspektive des Bee-Bots ein und passt diese laufend an, löst Aufgabe selbstständig mit „Trockenweiterstellen“.				
Negative Notiz:	Kind hat sich trotz identer Aufgabe wie Kind zuvor weder Lösung noch Vorgehensweise des Kindes zuvor (Perspektivenausgleich zu Bee-Bot) gemerkt, hat bei der Methode				

	„Trockenweiterstellen“ durch überhastetes Agieren immer wieder Probleme trotz Hilfestellung
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	154	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (selbstgewählter Anfangsposition – H2, Blick nach links, zu selbstgewähltem Endsymbol – A2, Blick nach links) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): VVVVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe selbstständig ohne „Trockenweiterstellen“.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	155	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition der Aufgabe zuvor – A2, Blick nach links, zu selbstgewähltem Endsymbol – H2, Blick nach rechts) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): LLVVVVVV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit kleinen Hilfestellungen.				
Negative Notiz:	Kind möchte den schwierigeren Teil der Aufgabe (180 Grad Drehung) zuerst händisch lösen, muss zum Einprogrammieren angehalten werden.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	156	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition der Aufgabe zuvor – H2, Blick nach rechts, zu selbstgewähltem Endsymbol – D1, Blick nach unten) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): LLVVVVLV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe mit „Trockenweiterstellen“ und kleiner Hilfestellung.				
Negative Notiz:	Hat nach Wegeinspeicherung Ausrichtung auf dem Startfeld vergessen - benötigt Hilfestellung vor Start der Abarbeitung der Befehle durch Drücken von „GO“.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	157	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition der Aufgabe zuvor – D1, Blick nach unten, zu selbstgewähltem Endsymbol – G3, Blick nach oben) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): RÜLVVVLV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein				

Aufgabensetting):	Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Kind löst Aufgabe mit „Trockenweiterstellen“ mit kleiner Hilfestellung.
Negative Notiz:	Kind spricht nicht bei der Wegplanung / Befehlseingabe.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	158	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von selbstgewählter Ausgangsposition – A2, Blick nach rechts, zu selbstgewähltem Endsymbol – H3, Blick nach oben) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): VVVVVVVLV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Ein Kind wählt ein Anfangs- und ein Endsymbol des Weges, den es mit dem Bee-Bot auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind löst Aufgabe bis auf anfänglichen Hinweis auf Clear ohne Hilfestellung.				
Negative Notiz:	Kind vergisst anfänglich auf Speicher löschen durch Clear – benötigt Hinweis.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	159	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition der Aufgabe zuvor – H3, Blick nach oben, zu selbstgewähltem Endsymbol – A3, Blick nach oben) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): RÜLVVVVVVRV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind rutscht selbst physisch Weg, den Bee-Bot fahren soll, ab. Konzentration trotz Unruhe anderer Kinder.				
Negative Notiz:	Kind benötigt viel Hilfestellung beim Lösen der Aufgabe mit „Trockenweiterstellen“, besonders bei Drehungen.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	160	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	A4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Selbstgewählte Einzelaufgabe (von Endposition der Aufgabe zuvor – A3, Blick nach oben, zu selbstgewähltem Endsymbol – E3, Blick nach oben) Kind im Sitzkreis auf Straßenmatte (8 mal 3 Felder): RRVLVVVVLV</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, Straßenmatte. Alle sitzen im Sitzkreis. Der Bee-Bot steht auf einem Symbol (Endposition der Aufgabe des Kindes davor), dies ist das neue Anfangssymbol. Ein Kind wählt das Endsymbol des Weges, den es mit dem Bee-Bot ausgehend vom Anfangssymbol auf der Matte befahren möchte. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.				
Positive Notiz:	Kind zeigt geplanten Weg für den Bee-Bot vor Einspeicherung auf der Matte an, löst Aufgabe mit Unterstützung mit „Trockenweiterstellen“. Kind umgeht durch anfängliche 180 Grad Drehung Notwendigkeit des Rückwärtsfahrens -> wusste für sich Befehlsplanung zu vereinfachen.				
Negative Notiz:	Kind vergisst anfänglich auf Speicher löschen durch Clear - benötigt Hinweis.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	34	Schwierigkeitsgrad:	4	Dauer:	5 Minuten
Aktivitätscluster:	A5) Gestelltes Problem lösen: Schleifen			Sozialform:	Einzel in ganzer Gruppe

Aufgabenart:	Programmieren
Aufgabe:	<i>Von TR gestellte Einzelaufgabe – Kind im Sitzkreis auf selbstgezeichneter Matte: Kind soll von der 4*4 Matte jede einzelne „Spalte“ nacheinander abfahren – startet bei Fahnen mit Blick nach oben, Ende bei Fischen mit Blick nach unten: VVVRVRVVVLVLRVVVRVVV</i>
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot, selbstgezeichnete Matte. Alle sitzen im Sitzkreis. TR gibt den Weg vor, den das Kind mit dem Bee-Bot auf der Matte befahren soll. In diesem Fall soll das Kind am linken unteren Feld der Matte beginnend nacheinander alle „Spalten“ der Matte abfahren. Das Kind programmiert den Weg ein, die anderen Kinder und TR und US unterstützen es dabei, falls erforderlich.
Positive Notiz:	Kind behilft sich bei Programmierung schwieriger Route an neuralgischen Punkten mit der Methode „Trockenweiterstellen“, löst fehler- und hilfefrei.
Negative Notiz:	Unruhe und Stören der zuschauenden Kinder.
Allgemeine Erkenntnis:	Aufgabe mit sehr hohem Schwierigkeitsgrad. Sehr unterschiedliches Leistungsniveau auch bedingt durch unterschiedliches Alter der Kinder.

AufgabenID:	202	Schwierigkeitsgrad:	4	Dauer:	15 Minuten
Aktivitätscluster:	A6) Selbstgewähltes Problem lösen: Schleifen			Sozialform:	Kleingruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Slalom fahren mit Bee-Bot um Hindernisse ohne Matte – immer zwei Kinder programmieren einen Bee-Bot und beraten sich bei den einzuspeichernden Befehlen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladene Bee-Bots in ausreichender Anzahl – mindestens einer für zwei Kinder, eventuell leere Dosen als mögliche Hindernisse – ansonsten können hier Gegenstände aus dem Kindergarten als Hindernisse herangezogen werden. Aufgabe für die Kinder ist es, mit dem Bee-Bot einen Slalomkurs um Hindernisse ohne eine Matte zu fahren, wobei immer zwei Kinder gemeinsam einen Bee-Bot programmieren und sich bei den einzuspeichernden Befehlen beraten. Dazu im Vorfeld durch TR und US mehrere Slalomkurse im Raum verteilt mit zuerst wenigen Hindernissen, später mit mehreren Hindernissen aufbauen, die vom Bee-Bot umfahren werden sollen. Die Übung kann in weiterer Folge so ausgebaut werden, dass zwei parallele Slalomkurse aufgebaut werden und die Kinder mit ihren Bee-Bots um die Wette fahren sollen.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Das Bearbeiten dieser Aufgabe führte bei den Kindern zu der Erkenntnis, dass ein größeres Repertoire an verschiedenen Befehlen beim Bee-Bot hilfreich wäre (-> der Bodenroboter Typ Probot würde diese anbieten), besonders das Wählen anderer Winkel wäre gut, da es mühsam ist rund um Hindernisse nur mit möglichen rechten Winkeln zu fahren und andere Schrittweiten (individuell wählbare) wären auch gut.				

AufgabenID:	203	Schwierigkeitsgrad:	4	Dauer:	15 Minuten
Aktivitätscluster:	A6) Selbstgewähltes Problem lösen: Schleifen			Sozialform:	Kleingruppe
Aufgabenart:	Programmieren				
Aufgabe:	<i>Mit dem Bee-Bot rund um ein Quadrat fahren ohne Matte (z.B. rund um einen quadratischen Polster)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladene Bee-Bots in ausreichender Anzahl – mindestens einer für zwei Kinder, quadratische Hindernisse in ausreichender Anzahl (Sitzpolster, Sitzteppich, Sessel, Hocker, etc.) – ein Hindernis pro Bee-Bot. Aufgabe ist es, mit dem Bee-Bot ohne Matte rund um ein Quadrat zu fahren (z.B. rund um einen quadratischen Polster) -> Kinder sollen den Bee-Bot so programmieren, dass dieser zuerst einmal, dann zweimal, dann dreimal in einer Fahrt rund um das quadratische Hindernis fährt. TR bzw. US geht zwischen den Kleingruppen, die im Raum verteilt an dieser Aufgabe arbeiten, herum und unterstützt mit Hinweisen und falls erforderlich mit Hilfestellungen.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	204	Schwierigkeitsgrad:	3	Dauer:	10 Minuten
Aktivitätscluster:	A6) Selbstgewähltes Problem lösen: Schleifen			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Reflexion der Aufgabe, den Bee-Bot um ein quadratisches Hindernis fahren zu lassen – heranzuführen an Wiederholungen, Schleifenkonzept (im Sitzkreis)</i>				
Beschreibung (Material und Aufgabensetting):	Keine besonderen Materialien benötigt. Alle sitzen im Sitzkreis. Besprechen der Erkenntnisse der Kinder beim Lösen der Aufgabe mit dem Bee-Bot um ein Quadrat zu fahren (ohne Matte), Heranzuführen mit Nachfragen bezüglich des Aufbaus des eingespeicherten Weges an die Begriffe „Wiederholungen“, „Schleifenkonzept“ bzw. auch „Algorithmusbegriff“. Hilfreiche Fragen in diesem Zusammenhang könnten sein: „Wie kann ich den Weg rund um ein Quadrat zerlegen?“, „Welche Teile des Weges sind gleich?“, „Wie schaut der Weg aus, wenn ich ein zweites / drittes / viertes				

	Mal rund um das Quadrat fahren möchte?".
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Voraussetzung: ID203 Besprechen der Erkenntnisse der Kinder beim Lösen der Aufgabe ID203.

A2.2 Aufgaben Inhaltsdimension B

AufgabenID:	113	Schwierigkeitsgrad:	4	Dauer:	20 Minuten
Aktivitätscluster:	B1) Verbalisierte elementare Verarbeitungsschritte		Sozialform:		Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen				
Aufgabe:	<i>Programmier-Roboter-Spiel – Kinder sollen sich in Zweier-Gruppen zusammenfinden, immer eine / einer der 2er Gruppe gibt der / dem anderen Anweisungen, was sie / er tun soll. Diese / dieser spielt Roboter und macht nur genau das, was angesagt wird.</i>				
Beschreibung (Material und Aufgabensetting):	Es werden keine besonderen Materialien benötigt. Die Kinder sollen sich in Zweier-Gruppen zusammenfinden, immer eine / einer der Zweier-Gruppe gibt der / dem anderen Anweisungen, was sie / er tun soll. Diese / dieser spielt Roboter und macht nur genau das, was angesagt wird. Immer eine 2er Gruppe arbeitet an dieser Aufgabenstellung, die anderen Kinder und TR und US sitzen im Halbkreis und unterstützen, falls notwendig. Bei der Erklärung der Aufgabe darauf eingehen, dass Roboter programmiert werden, damit die Kinder wissen was sie tun sollen. Unterschied zum Menschen, dessen Gehirn die einzelnen Körperfunktionen steuert, erklären – Bogen spannen zu dieser Aufgabe, wo in der Zweier-Gruppe das Kind, das die Anweisungen gibt, das andere Kind, das „Roboter“ spielt, bezüglich der Bewegungen programmiert.				
Positive Notiz:	Kinder sind sehr motiviert; Kinder nehmen Hilfestellungen an und setzen diese um.				
Negative Notiz:	Verwechslung von links / rechts bei Befehlsgabe, Befehlsangaben zu unpräzise (meist ohne Richtungsangabe), Unruhe bei Kindern, die nicht aktiv an der Aufgabe beteiligt sind.				
Allgemeine Erkenntnis:	Interessante abwechslungsreiche Übung. Sollte von TR und einer US eindeutig exemplarisch vorgezeigt werden um u.a. die Vorgehensweise und notwendige Genauigkeit beim Formulieren und Ausführen der Befehle zu verdeutlichen.				

AufgabenID:	50	Schwierigkeitsgrad:	3	Dauer:	5 Minuten
Aktivitätscluster:	B2) Gestelltes Problem lösen: visualisierte Sequenzen		Sozialform:		ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>TR spielt Roboter, US steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten (Richtungsbefehle, GO, Clear, Pause) und wiederholt mit den Kindern die Auswirkungen der einzelnen Befehle – Kärtchen: VVLVRVRüGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Halbkreis. TR spielt Roboter, US steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. US legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Darauf hinweisen, dass Karte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der, den die Befehlskarte beinhaltet. Bevor TR das aufgelegte Programm abarbeitet, hält TR symbolisch eine Clear-Karte vor sich um anzuzeigen, dass bisher „Eingespeichertes“, „vergessen“ wird und TR jetzt das von US aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem Go-Befehl beginnt abzarbeiten. Dieses Abarbeiten kann mit den Geräuschen, mit denen der reale Bee-Bot sein Abarbeiten untermalt, unterstützt werden.				
Positive Notiz:	Kinder arbeiten interessiert und aktiv mit, Verknüpfung mit bereits Gelerntem glückt, Kinder erkennen im Gespräch Unterschied in der Funktionalität zwischen Software und realem Bee-Bot.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	58	Schwierigkeitsgrad:	2	Dauer:	5 Minuten
Aktivitätscluster:	B2) Gestelltes Problem lösen: visualisierte Sequenzen		Sozialform:		ganze Gruppe
Aufgabenart:	Bewegen				
Aufgabe:	<i>Gemeinschaftsaufgabe: TR legt mit Kärtchen eine Befehlsfolge auf den Boden und alle Kinder und TR und US spielen gemeinsam die Roboter. Befehlsfolge: LRRL</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Bei dieser Aufgabe wird mit den Kindern gemeinsam an der Größe der Drehbewegungen gearbeitet. Dazu werden die Befehlskarten wie gewohnt am Boden aufgelegt, die Kinder sollen aufstehen und sich hintereinander in einer Reihe aufstellen – eine Armlänge entfernt vom Kind davor. TR steht vor den Kindern mit Blick in die gleiche Richtung wie diese, sagt die Befehle die abgearbeitet werden an und zeigt die Abarbeitung des Befehls vor.				

	Hebt zusätzlich auch immer den Arm der Richtung, in die gedreht werden soll hoch. Betont noch, dass es sich um eine Drehung handelt, man dabei also an derselben Stelle stehen bleibt und sich nur in eine andere Richtung dreht und zwar eine Vierteldrehung. Kinder, die einzelne Befehle falsch ausführen werden von US korrigiert.
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Links-Rechts-Probleme bei einzelnen Kindern, Größe der Drehbewegung teilweise nicht 90 Grad.
Allgemeine Erkenntnis:	Wenn Aufgabe ID58 und ID59 durchgeführt werden, dann mit Aufgabe ID59 beginnen. Nachdem bei den Aufgaben des Typs „Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert“ oftmals der „Kind-Roboter“ große Probleme mit der Größe der Drehbewegung hatte, bietet diese Aufgabe die Möglichkeit, gemeinsam mit allen Kindern an diesen Drehbewegungen zu arbeiten.

AufgabenID:	59	Schwierigkeitsgrad:	2	Dauer:	5 Minuten
Aktivitätscluster:	B2) Gestelltes Problem lösen: visualisierte Sequenzen		Sozialform:	ganze Gruppe	
Aufgabenart:	Bewegen				
Aufgabe:	<i>Gemeinschaftsaufgabe: TR legt mit Kärtchen eine Befehlsfolge auf den Boden und alle Kinder und TR und US spielen gemeinsam die Roboter. Befehlsfolge: LLLL</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Bei dieser Aufgabe wird mit den Kindern gemeinsam an der Größe der Drehbewegungen gearbeitet. Dazu werden die Befehlskarten wie gewohnt am Boden aufgelegt, die Kinder sollen aufstehen und sich hintereinander in einer Reihe aufstellen – eine Armlänge entfernt vom Kind davor. TR steht vor den Kindern mit Blick in die gleiche Richtung wie diese, sagt die Befehle die abgearbeitet werden an und zeigt die Abarbeitung des Befehls vor. Hebt zusätzlich auch immer den Arm der Richtung, in die gedreht werden soll hoch. Betont noch, dass es sich um eine Drehung handelt, man dabei also an derselben Stelle stehen bleibt und sich nur in eine andere Richtung dreht und zwar eine Vierteldrehung. Kinder, die einzelne Befehle falsch ausführen werden von US korrigiert.				
Positive Notiz:	Interessierte Mitarbeit, Vorwissen zu Teilen eines Ganzen.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	TR sagt immer an, wohin im Raum man sich durch die Drehung bewegt - wir drehen uns zur Tür, wir schauen zur Wand, etc. Außerdem zählt TR mit – einmal, zweimal, etc. Aufgabe wird wiederholt, damit jedes Kind merkt dass es sich um vier Vierteldrehungen in dieselbe Richtung handelt. „Bei diesem vier Mal drehen haben wir uns einmal um uns selbst gedreht“ sagt TR und zeigt die vier Drehungen nochmals ganz langsam Drehung für Drehung vor und sagt, dass der Bee-Bot bei einer Drehbewegung einen Viertelkreis macht.				

AufgabenID:	51	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen		Sozialform:	Kleingruppe in ganzer Gruppe	
Aufgabenart:	Bewegen				
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VRVVLrÜGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.				
Positive Notiz:	Zusammenarbeit zwischen steuerndem und ausführendem Kind funktioniert mit Unterstützung.				
Negative Notiz:	Kind interpretiert Rechtsdrehung als Drehung mit Vorwärtsschritt, Perspektivenproblem bei Vorwärtsschritt – Vorwärts von Ausgangsposition aus anstelle von aktueller Kind-Roboter-Position, Perspektivenproblem bei Linksdrehung – links im Raum anstelle links aus Kind-Roboter-Position.				
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.				

AufgabenID:	52	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
--------------------	----	----------------------------	---	---------------	-----------

Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen	Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen		
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VLRüRüGO</i>		
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.		
Positive Notiz:	Kind-Roboter geht Befehle ohne Fehler ab.		
Negative Notiz:	Steuerndes Kind verweigert danach in Kind-Roboter-Rolle zu schlüpfen.		
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.		

AufgabenID:	53	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen	Sozialform:	Kleingruppe in ganzer Gruppe		
Aufgabenart:	Bewegen				
Aktivität:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VVVRüRüGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.				
Positive Notiz:	Aktive Beteiligung von zuschauenden Kindern, gute Fehlerkorrektur von „Kind-Roboter“.				
Negative Notiz:	Probleme bei Richtungsinterpretation der Richtungspfeile auf den Befehlskarten durch steuerndes Kind.				
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.				

AufgabenID:	54	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen	Sozialform:	Kleingruppe in ganzer Gruppe		
Aufgabenart:	Bewegen				
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VLLRVGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu				

	sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.
Positive Notiz:	Kinder lassen sich nicht durch Probleme beim Lösen der Aufgabe entmutigen.
Negative Notiz:	Probleme bei Richtungsinterpretation der Richtungspfeile auf den Befehlskarten durch steuerndes Kind, Probleme beim „Kind-Roboter“ bei der Befehlsausführung – sowohl Links-Rechts-Probleme als auch Problem nur die Schritte entsprechend den aufgelegten Befehl
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.

AufgabenID:	55	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen				
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VLVRüRRRGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.				
Positive Notiz:	Guter Umgang mit Hilfestellungen, Geduld.				
Negative Notiz:	Steuerndes Kind legt optisch den Weg vor, den „Kind-Roboter“ gehen soll – mit Hinweis dass Befehle in einer Reihe von unten nach oben angeordnet werden sollen glückt Befehlsvorgabe durch das Kind, Richtungsproblem bei „Kind-Roboter“.				
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.				

AufgabenID:	56	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen				
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: LVRRLRüVVGLGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.				
Positive Notiz:	Keine Probleme beim Auflegen der Befehlskarten bei steuerndem Kind, „Kind-Roboter“ geht				

	Befehle ohne Fehler ab.
Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.

AufgabenID:	57	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	B3) Selbstgewähltes Problem lösen: visualisierte Sequenzen			Sozialform:	Kleingruppe in ganzer Gruppe
Aufgabenart:	Bewegen				
Aufgabe:	<i>Ein Kind spielt Roboter und wird mit Befehlskarten von einem anderen Kind gesteuert – vor der Gruppe die sich im Sitzkreis befindet mit Unterstützung US und TR; Befehlsfolge: VRüLRRLRüVGO</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Sitzkreis, ein Kind spielt Roboter, ein anderes Kind steuert diesen durch Auflegen („Einspeichern“) von Befehlskarten. Das steuernde Kind legt die Befehle in einer Reihe von unten nach oben auf. Unten liegt der zuerst eingespeicherte Befehl, der als erstes ausgeführt werden soll, dann der nächste und so weiter bis dass ganz oben der GO-Befehl liegt. Wenn nötig Kinder darauf hinweisen, dass die Befehlskarte immer so ausgerichtet werden soll, dass rechts oben der Bee-Bot abgebildet ist. Der Befehl, der in dieser Ausrichtung zu sehen ist, ist der den die Befehlskarte beinhaltet. Bevor der „Kind-Roboter“ das aufgelegte Programm abarbeitet, bekommt der „Kind-Roboter“ symbolisch eine Clear-Karte und soll diese vor sich halten um anzuzeigen, dass bisher „Eingespeichertes“ „vergessen“ wird und der „Kind-Roboter“ jetzt das vom steuernden Kind aufgelegte Programm eingespeichert bekommt und nach symbolischem Drücken der Befehlskarte mit dem GO-Befehl beginnt dieses abzuarbeiten. Abschließend bekommt der „Kind-Roboter“ nochmals die Clear-Karte, um anzuzeigen, dass dieser jetzt den Weg wieder „vergessen“ kann.				
Positive Notiz:	Gute Korrektur der Probleme bei Richtungsinterpretation der Richtungspfeile auf den Befehlskarten nach Erklärung durch TR durch steuerndes Kind, vorausschauende Hinweise zur Befehlsausführung von steuerndem Kind an „Kind-Roboter“.				
Negative Notiz:	Probleme bei Richtungsinterpretation der Richtungspfeile auf den Befehlskarten durch steuerndes Kind, Probleme des „Kind-Roboters“ bei Drehbewegungen – dreht sich 180 Grad anstelle von 90 Grad in die gewünschte Richtung.				
Allgemeine Erkenntnis:	Zuvor ID50 zur Veranschaulichung der Vorgehensweise beim Auflegen der Befehlskarten, beim Abarbeiten der Befehle etc. durchführen. Zum Auflockern „Vergessenskarte“ (Clear) „Kind-Roboter“ geben, „damit wir sichergehen, dass er vorher noch nichts weiß“ – wird mit großem Gelächter bei den Kindern angenommen und lockert die Übung auf. Auch nach Abschluss der Aufgabe wird zur Auflockerung wieder die „Vergessenskarte“ gereicht, die sich die „Kind-Roboter“ vor die Stirn halten, um anzuzeigen, dass sie diesen Weg wieder vergessen.				

AufgabenID:	38	Schwierigkeitsgrad:	3	Dauer:	5 Minuten
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Kurzbesprechung des Wortes „Programmieren“</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot. Alle sitzen im Sitzkreis. Nachdem der Begriff „Programmieren“ ausgehend von der Frage an die Kinder „Was ist programmieren?“ kurz besprochen wurde, werden in weiterer Folge noch die Wörter „Befehl“, „speichern“ etc. besprochen und in der Erklärung in Verbindung mit dem Einspeichern der Befehle durch Tastendruck beim Bee-Bot gebracht um die Begriffe mit den Erfahrungen der Kinder mit dem Bee-Bot zu verknüpfen.				
Positive Notiz:	Richtige Erklärung Kinder				
Negative Notiz:	Kinder werden schnell unruhig.				
Allgemeine Erkenntnis:	Kurz halten, Kinder werden sehr schnell unruhig.				

AufgabenID:	49	Schwierigkeitsgrad:	2	Dauer:	7 Minuten
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen				
Aufgabe:	<i>Wiederholung der Befehle, die dem Bee-Bot gespeichert werden können anhand von Befehlskarten gemeinsam mit den Kindern im Sitzhalbkreis.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Befehlskarten. Kinder sitzen im Halbkreis, TR steht vorne und hat alle verschiedenen Kärtchen gut sichtbar auf dem Boden vor den Kindern aufgelegt. Die Kärtchen haben Symbole und sind zusätzlich mit dem jeweiligen Befehl beschriftet. Außerdem ist der Bee-Bot abgebildet um anzuzeigen in welche Richtung der Bee-Bot blickt und dass z.B. der Pfeil der einen Befehlskarte einen Vorwärtsbefehl und der Pfeil der anderen Befehlskarte einen				

	Rückwärtsbefehl bedeuten soll. TR geht Kärtchen für Kärtchen mit den Kindern durch, wobei immer zuerst die Kinder nach ihrer Meinung zur Bedeutung des Befehlskärtchens gefragt werden.
Positive Notiz:	Interesse der Kinder, Kinder erkennen Befehle des Bee-Bots auf den Befehlskarten wieder.
Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	112	Schwierigkeitsgrad:	1	Dauer:	7 Minuten
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	Einzel
Aufgabenart:	Bewegen				
Aufgabe:	<i>Robotertheater – Kinder sollen sich im Raum bewegen und spielen, dass sie ein Roboter sind – ohne nähere Vorgaben außer dass es nicht zu laut bzw. unruhig etc. werden soll</i>				
Beschreibung (Material und Aufgabensetting):	Es werden keine besonderen Materialien benötigt. Die Kinder sollen sich frei im Raum bewegen und spielen, dass sie ein Roboter sind (Bewegungen, Laute, etc.) – ohne nähere Vorgaben, außer, dass es nicht zu laut werden soll und ohne Gewalt gespielt werden soll.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Wenig Kreativität bei den Robotertheater-Bewegungen, bald Unruhe.				
Allgemeine Erkenntnis:	Entweder als Zwischenübung zur Bewegung, wenn vorher viel gegessen wurde oder als Vorübung für andere Übungen wie z.B. ID113 gut geeignet.				

AufgabenID:	201	Schwierigkeitsgrad:	4	Dauer:	20 Minuten
Aktivitätscluster:	B4) Chaos vs. Systematik			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Gestalten				
Aufgabe:	<i>Einen Algorithmus zerlegen in Verarbeitungsschritte und diese auf Algorithmenkärtchen zeichnen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, (Moderations-)Kärtchen in ausreichender Anzahl (ca. 6 Stück pro Kind), Farbstifte in ausreichender Anzahl und in unterschiedlichen Farben, Spitzer, Radiergummi, Klarsichtfolien zum Sammeln der Kärtchen pro Kind für das Einkleben ins Forschungsheft. Alle sitzen im Sitzkreis. Es soll ein Algorithmus in Verarbeitungsschritte zerlegt und diese Verarbeitungsschritte in Form einer Art „Bildgeschichte“ jeweils auf ein eigenes Kärtchen gezeichnet werden. Die Kärtchen werden pro Kind in einer Klarsichtfolie gesammelt und diese von TR ins jeweilige Forschungsheft der Kinder eingeklebt.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Als Hintergrundgeschichte zum Verständnis für die Kinder kann erzählt werden, dass ein Außerirdischer auf die Erde kommt und die Kinder diesem Dinge aus ihrem täglichen Leben lernen sollen. Als Beispiele können genannt werden sich in der Früh anziehen, etwas essen, eine Stiege raufgehen, schlafen gehen, etc. Die Kinder sollen sich für ein Thema entscheiden, das sie dem Außerirdischen lernen möchten und dieses in möglichst elementare, visuell darstellbare Verarbeitungsschritte zerlegt auf die Kärtchen zeichnen.				

A2.3 Aufgaben Inhaltsdimension C

AufgabenID:	5	Schwierigkeitsgrad:	1	Dauer:	7 Minuten
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Verteilen der Forschungshefte und erklären der Funktion derselben inkl. Erklärung des Begriffs „Forschen“</i>				
Beschreibung (Material und Aufgabensetting):	Pro teilnehmendem Kind wird ein unliniertes Heft mit 20 Blatt, A4 oder Quart (hier müssen ausgeteilte A4 Blätter quer eingeklebt werden, das Heft wird durch das kleinere Format im positiven Sinn schneller voll) benötigt. Alle sitzen im Sitzkreis. TR bespricht den Begriff „Forschen“ kurz kindgerecht und knüpft mit Fragen an die Forschungsvorerfahrungen der Kinder an.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Forschungsheft bzw. ausgeteilte Materialien für die Kinder bis zum letzten Programmtag aufbewahren – am Ende des Programmtages spätestens immer absammeln, damit nichts verloren geht.				

AufgabenID:	7	Schwierigkeitsgrad:	2	Dauer:	15 Minuten
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel
Aufgabenart:	Gestalten				
Aufgabe:	<i>Zeichne einen oder mehrere Roboter auf ein extra Blatt</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: unlinierte Blätter A4, mehr als ein Blatt pro teilnehmendem Kind, Blei- und Buntstifte in ausreichender Anzahl und Farbauswahl, Spitzer und Radiergummis. Die Kinder sitzen im				

Aufgabensetting):	Sitzkreis, abwechselnd ein Kind mit Blick in den Kreis und eines aus dem Kreis um während dem Zeichnen die eigene Kreativität der Kinder zu fördern und Gruppenarbeiten zu erschweren.
Positive Notiz:	Kreativität, Mehrleistung der Kinder.
Negative Notiz:	Schnell Unruhe, viel Betreuungsaufwand.
Allgemeine Erkenntnis:	Hierzu werden unlinierte Blätter ausgeteilt, die anschließend ins Forschungsheft geklebt werden. Mit den Extrablättern kann ein Durchdrücken der Zeichnungen im Forschungsheft verhindert werden und bei mehreren Zeichnungen nur das „Best of“ ins Forschungsheft geklebt werden. Außerdem sitzen die Kinder im Sitzkreis abwechselnd mit Blick nach innen und nach außen, um eigene Ideen ohne Abzeichnen zu fördern.

AufgabenID:	8	Schwierigkeitsgrad:	3	Dauer:	15 Minuten
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Präsentieren				
Aufgabe:	<i>Gezeichnete/n Roboter in der Gruppe präsentieren und erklären</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Blätter mit den Roboterzeichnungen der Kinder. Die Kinder sitzen im Sitzkreis und präsentieren reihum ihre Zeichnungen, die anderen Kinder werden angehalten nachzufragen, was die gezeichneten Roboter können, welcher Art sie genau sind, warum die Kinder gewisse Details bei ihren Robotern gezeichnet haben, usw. Wenn Kinder nicht Nachfragen übernimmt dies TR.				
Positive Notiz:	Fantasie, Interesse, Wertschätzung der Lösungen durch andere Kinder, Kreativität.				
Negative Notiz:	Manche Kinder zeichnen anstelle von Robotern andere Motive wie Blumen / Wiesen / Herzen, gewalttätige Erzählinhalte, Präsentationsunwille einzelner, Geduld bei TR stark gefordert.				
Allgemeine Erkenntnis:	Ohne entsprechendes Nachfragen erzählen die meisten Kinder relativ wenig über die Funktionalität etc. ihrer Roboter sondern eher Geschichten. Falls das Nachfragen zur genauen Funktionalität, den Bestandteilen kein anderes Kind übernimmt, hier unterstützen.				

AufgabenID:	107	Schwierigkeitsgrad:	2	Dauer:	15 Minuten
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel
Aufgabenart:	Gestalten				
Aufgabe:	<i>Zeichne einen oder mehrere Roboter ins Forschungsheft</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Blei- und Buntstifte in ausreichender Anzahl und Farbauswahl, Spitzer und Radiergummi. Die Kinder sitzen im Sitzkreis, abwechselnd ein Kind mit Blick in den Kreis und eines aus dem Kreis, um während dem Zeichnen die eigene Kreativität der Kinder zu fördern und Gruppenarbeiten zu erschweren.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Durchbrechen des Sitz-Aufgabensettings führte zu abzeichnen, direktes Zeichnen ins Forschungsheft führte zu ungeordneter Heftgestaltung und verstreuten Zeichnungen innerhalb des Forschungshefts.				
Allgemeine Erkenntnis:	Kinder werden im Sitzkreis so gesetzt, dass abwechselnd immer ein Kind nach innen und eines nach außen schaut – wird mit der Begründung gemacht, dass sich Kinder ja vielleicht überraschen lassen möchten was sich die anderen für Roboter ausgedacht haben.				

AufgabenID:	131	Schwierigkeitsgrad:	3	Dauer:	10 Minuten
Aktivitätscluster:	C1) Einführung Wissensarbeit			Sozialform:	Einzel in ganzer Gruppe
Aufgabenart:	Präsentieren				
Aufgabe:	<i>Kinder erklären / präsentieren Zeichnungen auf den selbstbemalten / -gezeichneten Matten</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: selbstgezeichnete Matten. Alle sitzen im Sitzkreis. Die Matten liegen in der Mitte des Sitzkreises am Boden und die Kinder kommen der Reihe nach dran und erklären / präsentieren was sie auf der Matte gezeichnet haben				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	1	Schwierigkeitsgrad:	1	Dauer:	15 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	ganze Gruppe
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Namensschilder erstellen und verteilen inklusive Vorstellungsrunde</i>				
Beschreibung (Material und Aufgabensetting):	Für jedes Kind und jede an den Lehr-Lern-Einheiten teilnehmende Betreuungsperson ein Namensschild – am besten zum Umhängen (wie in Unternehmen), Stifte zum Beschriften. Alle sitzen im Sitzkreis auf Sitzpolstern oder Sitzteppichen (gibt es im Kindergarten), TR und Betreuungspersonal stellen sich vor, hängen sich Namensschild um und fragen dann der Reihe nach die Kinder nach ihren Namen und erstellen Namensschilder, die sich die Kinder dann umhängen können.				

Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Klebeetiketten nicht geeignet, Ablenkung Kinder, Widerstand gegen die Aufgabe.
Allgemeine Erkenntnis:	Kinder im Kindergartenalter können in der Regel weder lesen noch schreiben, einige können ihren eigenen Namen schreiben. Der Hinweis, dass die Aufkleber keine Kleberückstände auf der Kleidung hinterlassen half um die Vorbehalte der Kinder abzubauen. Besser geeignet sind nach dieser Erfahrung aber sicher Namensschilder (wie in Unternehmen) zum Umhängen. Die Namensschilder sollten spätestens am Ende jedes Programmtages abgesammelt werden, damit diese nicht verloren gehen.

AufgabenID:	6	Schwierigkeitsgrad:	1	Dauer:	4 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Forschungsheft beschriften</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Blei- und Buntstifte, Spitzer und Radiergummi. Die Kinder beschriften soweit sie es schon können selbst mit Vornamen oder Symbol das Forschungsheft, TR hilft dabei.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Nur Blei- und Buntstifte für alle Kindergartenkinder erlaubt, manche Kinder malen im Heft statt Beschriftung.				
Allgemeine Erkenntnis:	Kinder beschriften mit Symbolen (wie im Kindergarten für Garderobenhaken, etc. gebräuchlich) oder Namen. Die meisten Kinder benötigen Hilfe beim Schreiben des Namens.				

AufgabenID:	9	Schwierigkeitsgrad:	1	Dauer:	5 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Einkleben der Roboterzeichnung(en) ins Forschungsheft</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte (Forschungsheft bzw. ausgeteilte Materialien für die Kinder bis zum letzten Programmtag aufbewahren - am Ende des Programmtages spätestens immer absammeln), Roboterzeichnungen der Kinder und Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Anzahl. Alle sitzen im Sitzkreis.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Arbeitsmittelplanungsfehler, Unruhe durch Wartezeit.				
Allgemeine Erkenntnis:	Kinder benötigen Hilfe beim Einkleben. Forschungshefte während des Programms immer wieder absammeln und zum nächsten Termin mitnehmen bzw. auch absammeln, wenn am selben Tag zu verschiedenen Zeitpunkten mit dem Forschungsheft gearbeitet wird um einen geordneten Ablauf zu gewährleisten.				

AufgabenID:	20	Schwierigkeitsgrad:	3	Dauer:	2 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aktivität:	<i>Foto eines Staubsaugerroboters erstellen.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Kinderkameras, Staubsaugerroboter. Die Aufgabe ist es, ein Foto vom Staubsaugerroboter ohne Bewegung desselben zu machen.				
Positive Notiz:	Kinder haben Freude am Fotografieren.				
Negative Notiz:	Eigentliche Aufgabenstellung für Kinder tendenziell uninteressant.				
Allgemeine Erkenntnis:	Eher kurze Zeitspanne für die Aufgabe wählen um das Auffinden der Zusatzfunktionalitäten der Kinderkamera, wie Spiele, zu umgehen.				

AufgabenID:	23	Schwierigkeitsgrad:	3	Dauer:	8 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Vorführung eines Staubsaugerroboters und Aufgabe den Staubsaugerroboter mit den Kinderkameras in Fahrt zu fotografieren.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Staubsaugerroboter, entweder für jedes teilnehmende Kind eine Kinderkamera, oder immer für zwei Kinder eine Kinderkamera (jeweils aufgeladen). Zuerst wird der Staubsaugerroboter in Betrieb genommen – auf genügend Platz achten, die Kinder dürfen aufstehen und den fahrenden Roboter nachgehen, zuschauen etc. Dann soll jedes Kind mit der Kinderkamera ein Foto des fahrenden Staubsaugerroboters machen.				
Positive Notiz:	Erweiterter Funktionsumfang der Kamera für Kinder durch Ausprobieren erfassbar.				
Negative Notiz:	Berührungssängste mit fahrendem Roboter.				
Allgemeine Erkenntnis:	Kinder versuchten den Roboter zuerst im Sitzen zu fotografieren und benötigten Hilfestellung, dass das Erstellen von Fotos einfacher gelingt, wenn sie dazu aufstehen, dem Roboter im Raum folgen.				

AufgabenID:	24	Schwierigkeitsgrad:	1	Dauer:	10 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Bisher „erforschte“ Roboter im Forschungsheft dokumentieren mit Bild bzw. selbstgemachtem Foto (des Staubsaugerroboters)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Bild vom Bee-Bot für jedes teilnehmende Kind, ausgedruckte Fotos des Staubsaugerroboters (das Ausdrucken der Fotos aus Zeitgründen außerhalb des Programms organisieren), Klebstoff, der im Kindergarten verwendet werden darf, in ausreichender Anzahl, Kindergarten-adäquate Scheren in ausreichender Anzahl. Alle sitzen im Sitzkreis. Die Kinder bekommen ihr Forschungsheft, ein Bild des Bee-Bots und ein Foto des Staubsaugerroboters und sollen diese in ihr Forschungsheft (auf Seiten unmittelbar hintereinander) einkleben. Sie dürfen auch z.B. den Bee-Bot entlang der Konturen ausschneiden.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Teilweise fehlende Forschungshefte durch nicht erfolgte Materialverwaltung, Unflexibilität bei Arbeitsmitteln.				
Allgemeine Erkenntnis:	Ganz genaue Anweisungen geben – wohin sollen die Bilder geklebt werden, was soll ausgeschnitten werden, etc. Die Kinder benötigen Unterstützung beim Ausschneiden und Einkleben – mehrere Betreuungspersonen notwendig.				

AufgabenID:	111	Schwierigkeitsgrad:	1	Dauer:	7 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Einkleben der Bilder der Roboter ins Forschungsheft mit Unterstützung</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, schon ausgeteilte Bilder der erklärten Roboter, Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Menge. Alle sitzen im Sitzkreis. Die Kinder sollen die Roboterbilder, die sie bekommen haben, im Forschungsheft einkleben.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Erklärung notwendig, wie mit den unterschiedlichen Bildgrößen beim Einkleben umgegangen werden soll bzw. dass mehrere Bilder, wenn dies von der Größe her möglich ist, auf eine Seite geklebt werden sollen, direkt im Anschluß an das zuletzt bemalte / beklebte Blatt mit dem Einkleben der neuen Bilder begonnen werden soll. Unterstützung notwendig beim Einkleben. Forschungsheft nach dem Einklebevorgang einsammeln.				

AufgabenID:	142	Schwierigkeitsgrad:	3	Dauer:	10 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Erstellen von Fotos des fahrenden Staubsaugerroboters mit soeben ausgeteilten Kinderkameras.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: entweder für jedes teilnehmende Kind eine Kinderkamera, oder immer für zwei Kinder eine Kinderkamera (jeweils aufgeladen), aufgeladener Staubsaugerroboter. Jedes Kind (oder Zweier-Gruppe) bekommt eine Kinderkamera – Unterstützung beim Einschalten, Bedienen bei manchen Kindern nötig. Staubsaugerroboter bewegt sich im Raum, Kinder werden angehalten sich auch im Raum zum Staubsaugerroboter zu bewegen um das Fotografieren für sie zu erleichtern.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Durch die kurze Zeitspanne, die zur Verfügung gestellt wird, kann das Auffinden der Spiele auf der Kamera und die Ablenkung durch diese gedämmt werden. Den Kindern bleibt nach Erstellen des / der Fotos des fahrenden Staubsaugerroboters wenig Zeit für andere Funktionalitäten der Kamera. Nach Beendigung des Fotografierens werden die Fotoapparate wieder eingesammelt und weggeräumt. Außentaschen der Fotoapparate sollen mit Namensaufkleber versehen werden, damit ein Ausdrucken der individuellen Fotos der Kinder bis spätestens zum letzten Programmtag stattfinden kann.				

AufgabenID:	143	Schwierigkeitsgrad:	1	Dauer:	6 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	Einzel
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Einkleben eines Bilds des Bee-Bots ins Forschungsheft.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Bilder des Bee-Bots in ausreichender Anzahl, Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Anzahl. Alle sitzen im Sitzkreis. Kinder sollen das ihnen ausgeteilte Bild des Bee-Bots ins Forschungsheft einkleben. Es wird gezeigt wo sie das Bild des Bee-Bots im Forschungsheft einkleben sollen (Chronologie).				
Positive Notiz:	Keine positiven Auffälligkeiten.				

Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Unterstützung beim Einkleben. Anschließend wieder einsammeln der Forschungshefte, wenn im Anschluss keine weiteren Aufgaben mit dem Forschungsheft gemacht werden.

AufgabenID:	145	Schwierigkeitsgrad:	4	Dauer:	15 Minuten
Aktivitätscluster:	C2) Forschungsheft			Sozialform:	ganze Gruppe
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Besprechen der Chronologie der Ereignisse in den Lehr-Lern-Einheiten anhand der Einträge im Forschungsheft im Sitzkreis und Erarbeitung Datumsbegriff sowie Beschriftungsnotwendigkeit von Forschungsergebnissen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Bunt- und Filzstifte für TR und US um Datumseinträge in den Forschungsheften der Kinder vornehmen zu können. Alle sitzen im Sitzkreis. Gemeinsames Erarbeiten der Begriffe Jahr, Monat, Tag und besprechen, dass die Kombination aus diesen dreien das eindeutige Datum z.B. des heutigen Tages angibt.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Das Datum selbst zu schreiben ist für die Kinder noch zu schwierig – daher die Beschriftung durch TR und US vornehmen und in den von den Kindern gewünschten Farben die jeweiligen Einträge in deren Forschungsheft mit dem entsprechenden Datum versehen.				

AufgabenID:	26	Schwierigkeitsgrad:	2	Dauer:	25 Minuten
Aktivitätscluster:	C3) Konzeption und Erstellung Lernmaterial			Sozialform:	Kleingruppe
Aufgabenart:	Gestalten				
Aufgabe:	<i>Eigene Matten zeichnen – Kinder in Kleingruppen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Blätter mit einem vorgezeichneten Raster in Schrittweite des Bee-Bots je Kleingruppe von drei Kindern mindestens ein Blatt (z.B. 5 mal 4 Kästchen auf Flipchart- oder Backpapier), Farbstiftsets, Spitzer und Radiergummi für jede Kleingruppe, wiederablösbares Klebeband um die Mattenrohlinge am Boden festzukleben. Die Kinder sollen diese Mattenrohlinge in Kleingruppen ähnlich zu den schon bespielten gekauften Matten mit eigenen Ideen gestalten.				
Positive Notiz:	Beteiligung aller Kinder, Diskussion untereinander.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Ausreichender Platz rund um die jeweilige Zeichnung (am Boden – die Kinder malen gern am Boden auf Polstern rund um die Matte sitzend) wird benötigt, um zu verhindern, dass die Kinder beim Holen von Stiften, Wechseln der Mattenseite, etc. auf die Matte steigen oder gar die Ecken umknicken / einreißen etc.				

AufgabenID:	48	Schwierigkeitsgrad:	4	Dauer:	25 Minuten
Aktivitätscluster:	C3) Konzeption und Erstellung Lernmaterial			Sozialform:	Einzel
Aufgabenart:	Gestalten				
Aufgabe:	<i>Erstellen eines eigenen Prototyps eines Roboters, mit dem ein anderes Kind Programmieren lernen kann.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Bastelmaterialien in ausreichender Menge: A4 Blatt Papier für jedes Kind, zusätzliches Papier, Moosgummi in verschiedenen Farben, Stifte, Schere, Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Menge; Forschungsheft, Klarsichthüllen zur späteren Aufbewahrung / Sammlung. Aufgabenstellung: auf dem ausgeteilten losen Blatt zuerst überlegen wie der Roboter aussehen kann, dann diese Vorlage ausschneiden, auf den Moosgummi legen, abpausen und aus dem Moosgummi nach der Vorlage ausschneiden. Bedienelemente aus Moosgummi extra ausschneiden und dann an der gewünschten Position auf die Moosgummi-Grundform kleben. Hilfestellung: aus der Vogelperspektive anschauen und aus dieser Perspektive zeichnen – veranschaulichen anhand der Erarbeitung eines Musterprototyps des Bee-Bot was das bedeutet und wie dieser als Prototyp aussehen könnte. Nach genauer Erklärung der Aufgabe durch TR (ca. 5 Minuten) empfiehlt sich eine durchgehende Arbeitsdauer für die Kinder von ca. 20 Minuten, danach sinkt die Aufmerksamkeit deutlich. Sollte noch mehr Zeit benötigt werden empfiehlt sich eine Weiterarbeit zu einem anderen Zeitpunkt. Für jedes Kind die jeweilige Vorlagenzeichnung und den Roboter-Prototyp in eine Klarsichthülle geben, die ins Forschungsheft eingeklebt werden kann - absammeln.				
Positive Notiz:	Kinder sind kreativ, verwandeln ihnen bekannte Objekte in Bewegung (Tiere, Fahrzeuge, Gliedmaßen...) in Roboter; Kinder abstrahieren / reduzieren Funktionalitäten und konzentrieren sich auf die Figuren.				
Negative Notiz:	Optische Reduktion von Funktionalitäten und Erzählung der Funktionalitäten (=> Viele Funktionalitäten nicht berücksichtigt) Sinkende Aufmerksamkeit bei Warten auf Unterstützung durch TR. Kopien von Prototypen anderer Kinder. Gelegentlich werden Roboter mit „bösen“ Wesen gleichgesetzt. Aufgabenstellung für Kinder stellenweise zu abstrakt.				

Allgemeine Erkenntnis:	Hinweis auf die geeignete Größe des Prototyps, damit ein Kind mit diesem Roboter auch ordentlich hantieren könnte. Aufgabe mit hohem Schwierigkeitsgrad, die erst gegen Ende des Programms angesetzt werden sollte. Kinder benötigen sehr hohes Maß an Unterstützung und Hilfe – und fordern ungeteilte Aufmerksamkeit von den TR und US. Benötigen Unterstützung mit den Fragen: wie viele Tasten soll der Roboter zur Bedienung bekommen, wie soll er bedient werden? Diese Aufgabe ist für eine altersgemischte Kindergartengruppe (3 bis 6 Jahre) zu schwer in fast allen ihren Teilen - > für 6-jährige Kindergartenkinder (Vorschulkinder) bei geeigneter Vorarbeit geeignet, da hier das Unterstützen beim Ausschneiden, Kleben, Zeichnen an sich wegfällt und sich die Unterstützung auf Prototyp-bezogene Fragestellungen beschränkt. Hilfestellung: Zuerst überlegen: ist der Roboter quadratisch, dreieckig, rund oder ist er vielleicht ein Tier?
-------------------------------	---

AufgabenID:	13	Schwierigkeitsgrad:	2	Dauer:	5 Minuten
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen				
Aufgabe:	<i>Gemeinsame Wiederholung der Funktionsweise des Bee-Bot (Anknüpfen an den Vortag)</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot. Alle sitzen im Sitzkreis.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	14	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen				
Aufgabe:	<i>Unterscheidung Clear/Pause wird geschult</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Bee-Bot. Alle sitzen im Sitzkreis.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Clear- und Pause-Taste optisch für Kind gleich.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	18	Schwierigkeitsgrad:	3	Dauer:	7 Minuten
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen				
Aufgabe:	<i>Reflexionsrunde über freies Üben mit Bee-Bot</i>				
Beschreibung (Material und Aufgabensetting):	Alle sitzen im Sitzkreis. Kinder berichten reihum von ihren Erlebnissen mit dem Bee-Bot auf den vorgefertigten Matten.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	144	Schwierigkeitsgrad:	3	Dauer:	6 Minuten
Aktivitätscluster:	C4) Festigung des Wissens und Reflexion			Sozialform:	ganze Gruppe
Aufgabenart:	Wiederholen				
Aufgabe:	<i>Reflexion Vorführung Staubsaugerroboter und einkleben eines Fotos desselben ins Forschungsheft.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, (von den Kindern erstellte) Fotos des Staubsaugerroboters (entweder personalisiert, falls die Kameras mit Namensaufklebern versehen wurden und von jedem Kind / jeder Zweier-Gruppe Fotos ausgedruckt wurden – oder ein beliebiges Foto des Staubsaugerroboters in Fahrt, für jedes Kind das gleiche, falls dies nicht erfolgt ist), Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Anzahl. Alle sitzen im Sitzkreis, Kinder sollen das ihnen ausgeteilte Bild des Staubsaugerroboters ins Forschungsheft einkleben – es wird gezeigt, wo die Kinder das Bild des Staubsaugerroboters ins Forschungsheft einkleben sollen (Chronologie). TR regt die Kinder durch Nachfragen an, über ihre Erlebnisse beim Fotografieren des Staubsaugerroboters zu berichten.				
Positive Notiz:	Keine positiven Auffälligkeiten.				

Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Unterstützung beim Einkleben wenn nötig. Anschließend einsammeln der Forschungshefte, wenn im Anschluss keine weiteren Aufgaben mit dem Forschungsheft gemacht werden.

A2.4 Aufgaben Inhaltsdimension D

AufgabenID:	22	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Erklärung der Funktionsweise eines Staubsaugerroboters im Sitzkreis</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Staubsaugerroboter. Alle sitzen im Sitzkreis, TR erklärt die Funktionsweise des Staubsaugerroboters im ausgeschalteten Zustand, spricht über Sensoren, etc.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	25	Schwierigkeitsgrad:	1	Dauer:	15 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Vorzeigen und erklären weiterer Roboter anhand von Bildern – Kinder dürfen sich je zwei Bilder aussuchen und diese ins Forschungsheft einkleben</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Forschungshefte, Bilder von unterschiedlichen Robotern in ausreichender Anzahl (jedes Bild muss von allen Kindern gewählt werden können). Geeignet sind Bilder von unterschiedlichen Haushaltsrobotern, Industrierobotern, Marsrobotern, Rettungsrobotern, Roboterautos, Robotern zur Bombenentschärfung und auch Robotern, die in aktuellen, Kind adäquaten Filmen vorkommen, z.B. Wolly, Klebstoff, der im Kindergarten verwendet werden darf in ausreichender Menge. Alle sitzen im Sitzkreis. TR zeigt und erklärt die Roboter auf den Bildern, von denen sich die Kinder je zwei Bilder aussuchen dürfen. Die Kinder sollen diese Bilder ins Forschungsheft einkleben.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Unterstützung beim Einklebevorgang sofern erforderlich.				

AufgabenID:	109	Schwierigkeitsgrad:	1	Dauer:	10 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Vorzeigen und erklären weiterer Roboter anhand von Bildern in der Gruppe durch TR</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Bilder von unterschiedlichen Robotern. Geeignet sind Bilder von unterschiedlichen Haushaltsrobotern, Industrierobotern, Marsrobotern, Rettungsrobotern, Roboterautos, von Robotern zur Bombenentschärfung und auch Robotern, die in aktuellen, Kind adäquaten Filmen vorkommen, z.B. Wolly. Alle sitzen im Sitzkreis, TR zeigt und erklärt die Roboter auf den Bildern.				
Positive Notiz:	Interesse, alle Kinder aktiv, Kinder die den Film mit Wolly dem Roboter gesehen haben erzählen die Handlung des Films, Kreativität.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	110	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Bilder aller gezeigten Roboter werden an die Kinder ausgeteilt</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Bilder der von TR gezeigten und erklärten Roboter in ausreichender Anzahl. Jedes im Sitzkreis sitzende Kind bekommt von jedem gezeigten Roboter ein Bild.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Automatische Verteilung von Materialien von einem Kind zum nächsten funktionierte hier nicht.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	141	Schwierigkeitsgrad:	2	Dauer:	7 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Erklärung der Funktionsweise eines Staubsaugerroboters im Sitzkreis und Vorführung des Staubsaugerroboters.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: aufgeladener Staubsaugerroboter. Alle sitzen im Sitzkreis, TR erklärt die Funktionsweise des Staubsaugerroboters im ausgeschalteten Zustand, spricht über Sensoren, etc. Dann wird der Staubsaugerroboter in Betrieb genommen, wobei auf genügend Platz für die Fahrt zu achten ist. Die Kinder dürfen aufstehen und dem fahrenden Roboter nachgehen, zuschauen etc.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	161	Schwierigkeitsgrad:	2	Dauer:	6 Minuten
Aktivitätscluster:	D1) Technik im Alltagskontext			Sozialform:	ganze Gruppe
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Erklärung Funktion Fotodrucker mit Einbindung der Kinder im Sitzkreis</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Fotodrucker, der ohne Verbindung zum Computer direkt von Speicherkarte oder durch Verbindung zur Kamera mit USB-Kabel Fotos drucken kann, ausreichend Fotopapier für diesen Drucker, vergewissern dass Patronen genügend gefüllt sind. Kinder sitzen im Sitzkreis. TR erklärt mit Einbindung der Kinder die Funktion des Fotodruckers.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Hier kann eine Verbindung zum Bee-Bot, zu anderen bereits gesehenen Robotern, Kinderkameras geschaffen werden durch Hinweis auf Funktionstasten, Farben, etc.				

AufgabenID:	2	Schwierigkeitsgrad:	3	Dauer:	5 Minuten
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Habt ihr schon einmal einen Roboter gesehen? Etwas über Roboter gehört?</i>				
Beschreibung (Material und Aufgabensetting):	Keine besonderen Materialien erforderlich. Alle sitzen im Sitzkreis. TR stellt die Frage und regt Kinder zum Mitarbeiten an.				
Positive Notiz:	Interesse der Kinder, bringen Beispiele aus ihrem Alltag.				
Negative Notiz:	Gesprächsregeln können nicht vorausgesetzt werden, Kind mit eigenem Roboter zeigt sich nicht kooperativ bezüglich Informationen darüber für die anderen Kinder.				
Allgemeine Erkenntnis:	Entweder ganz zu Beginn oder an dieser Stelle müssen Gesprächsregeln aufgestellt werden, da der Lärmpegel ansonsten steigt und alle Kinder durcheinander rufen.				

AufgabenID:	3	Schwierigkeitsgrad:	3	Dauer:	7 Minuten
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Lebt ein Roboter?</i>				
Beschreibung (Material und Aufgabensetting):	Keine besonderen Materialien erforderlich. Alle sitzen im Sitzkreis. TR stellt die Frage und regt Kinder zum Mitarbeiten an, fragt nach zu Nahrung, Sinnesorganen und Wahrnehmung von Robotern.				
Positive Notiz:	Interesse, Beteiligung				
Negative Notiz:	Gesprächsregeln können nicht vorausgesetzt werden.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	4	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer			Sozialform:	ganze Gruppe
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Wozu kann man Roboter verwenden?</i>				
Beschreibung (Material und Aufgabensetting):	Keine besondere Materialien erforderlich. Alle sitzen im Sitzkreis. TR stellt die Frage und regt Kinder zum Mitarbeiten an.				
Positive Notiz:	Vorwissen				

Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Keine Anmerkungen.

AufgabenID:	102	Schwierigkeitsgrad:	4	Dauer:	5 Minuten
Aktivitätscluster:	D2) Vorwissen und Wissenstransfer		Sozialform: ganze Gruppe		
Aufgabenart:	Fragen entwickelnd unterrichten				
Aufgabe:	<i>Gruppengespräch über Computer</i>				
Beschreibung (Material und Aufgabensetting):	Es werden keine besonderen Materialien benötigt. Alle sitzen im Sitzkreis. TR moderiert ein Gruppengespräch über Computer. Es werden Unterschiede zu anderen Gegenständen des täglichen Gebrauchs, wie etwa Ofen, etc. oder Menschen, etc., herausgearbeitet. In groben Zügen wird die Funktionsweise von Computern besprochen.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Diese Aufgabe eignet sich unter anderem gut dazu, die Vorerfahrungen der Kinder mit Computern in Erfahrung zu bringen.				

AufgabenID:	19	Schwierigkeitsgrad:	2	Dauer:	4 Minuten
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte		Sozialform: Einzel		
Aufgabenart:	Erforschen				
Aufgabe:	<i>Vertraut machen mit Kinderkameras</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: entweder für jedes teilnehmende Kind oder immer für zwei Kinder eine Kinderkamera (jeweils aufgeladen). Kinder dürfen die Kinderkameras frei erkunden und ausprobieren.				
Positive Notiz:	Begeisterung über Kamera.				
Negative Notiz:	Ablenkung durch Spiele auf Kameras.				
Allgemeine Erkenntnis:	Am Ende jedes Programmtages darauf achten, dass die Kinder die Kinderkameras nicht irrtümlich mitnehmen und spätestens am Ende des Programmtages absammeln. Eher kurze Zeitspanne für die Aufgabe wählen um das Auffinden der Zusatzfunktionalitäten wie Spiele zu umgehen.				

AufgabenID:	21	Schwierigkeitsgrad:	2	Dauer:	2 Minuten
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte		Sozialform: Einzel		
Aufgabenart:	Erforschen				
Aufgabe:	<i>Wer kann am schnellsten den Fotoapparat ausschalten?</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Kinderkameras. Zwischenspiel um die Ablenkung der Kinder während der Erklärung des Staubsaugerroboters durch die Kamera zu verhindern				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	162	Schwierigkeitsgrad:	3	Dauer:	16 Minuten
Aktivitätscluster:	D3) Exploration technischer Alltagsgeräte		Sozialform: Einzel in ganzer Gruppe		
Aufgabenart:	Dokumentieren				
Aufgabe:	<i>Ausdrucken von Fotos durch die Kinder mit Unterstützung von TR.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Fotodrucker, der ohne Verbindung zum Computer direkt von Speicherkarte oder durch Verbindung zur Kamera mit USB-Kabel Fotos drucken kann, ausreichend Fotopapier für diesen Drucker, vergewissern dass Patronen genügend gefüllt sind. Alle sitzen im Sitzkreis. Jedes Kind darf reihum ein Foto ausdrucken.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Das Funktionsprinzip des verwendeten Druckers kann hier während des Ausdruckens erklärt werden (z.B. wenn das Foto immer wieder eingezogen wird und sich die Farben des Fotos langsam aufbauen).				

A2.5 Aufgaben Inhaltsdimension E

AufgabenID:	403	Schwierigkeitsgrad:	4	Dauer:	4 Minuten
Aktivitätscluster:	E1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Vertraut machen mit der Steuerung des Bee-Bots in der Software Focus on Bee-Bot</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gefragt „Wie kann der Bienenroboter in der Software gesteuert werden?“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	<p>So soll erreicht werden, dass sich das Kind selbst über die Bedienung Gedanken macht und an schon Bekanntes aus der Bedienung anderer Spiele oder auch aus der Steuerung des realen Bee-Bots, falls es diesen schon kennengelernt hat, anknüpft und nicht passiv, ohne Verknüpfung mit Vorwissen des Kindes, eine Einführung in der Bedienung der Software durch TR bekommt.</p> <p>Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1 – Unterschiede zur Hardware: GO-Befehl ist grüner Pfeil in Symbolleiste, Links- und Rechts-Dreh-Buttons sind nicht gedreht, Clear ist rotes X rechts statt blaues Clear links, bei Clear wird der Bee-Bot an den Start zurück gestellt, bei Drücken des GO-Buttons wird der Bee-Bot wieder an die Startposition zurückgestellt und fährt von dort die gespeicherten Befehle ab (Kind kann bei der Software nicht wie beim realen Bee-Bot Teilstrecken einspeichern, diese mit GO ausführen, den Speicher leeren u dann die nächste Teilstrecke einspeichern um ans Ziel zu kommen).</p> <p>Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 2 – Unterschiede zur Hardware: es gibt zweites GO in der Symbolleiste, bei Clear wird der Bee-Bot an den Start zurückgestellt, bei Drücken des GO-Buttons wird der Bee-Bot wieder an die Startposition zurückgestellt und fährt von dort die gespeicherten Befehle ab (Kind kann bei der Software nicht wie beim realen Bee-Bot Teilstrecken einspeichern, diese mit GO ausführen, den Speicher leeren und dann die nächste Teilstrecke einspeichern um ans Ziel zu kommen).</p>				

AufgabenID:	409	Schwierigkeitsgrad:	4	Dauer:	2 Minuten
Aktivitätscluster:	E1) Exploration Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Speicher des Bee-Bots in der Software Focus on Bee-Bot löschen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot 1, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot 1 und wird gefragt: „Wie kann die Biene hier im Programm ‚vergessen‘? Bitte drücke vergessen und stelle die Biene wieder an den Start.“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	<p>Um die Funktionalität des Clear-Buttons ausprobieren zu können, empfiehlt es sich, dass bereits mindestens ein Befehl im Speicher gespeichert ist. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1: Der Clear-Button ist im Programm Focus on Bee-Bot 1 anders gestaltet als beim Hardware-Bee-Bot -> erkennt das Kind diese trotzdem? Der Clear-Button ist im Programm anders gestaltet als am realen Bee-Bot (rotes X rechts unten versus blauer Clear-Button links unten).</p> <p>Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1 und 2: Das Kind kann bei dieser Aufgabe das unterschiedliche Verhalten des Bee-Bots im Programm und des Hardware-Bee-Bots beobachten, da in der Software nach Drücken des Clear-Buttons der Bee-Bot automatisch an den Start zurück gestellt wird, was in der Realität mit dem Hardware-Bee-Bot nach Drücken der Clear-Taste nicht passiert, außerdem ist der Button im Programm ausgegraut und nicht klickbar dargestellt, solange kein Befehl eingespeichert wurde.</p>				

AufgabenID:	417	Schwierigkeitsgrad:	1	Dauer:	5 Minuten
Aktivitätscluster:	E2) Vorstellung Programmierumgebung			Sozialform:	Einzel
Aufgabenart:	Demonstrieren				
Aufgabe:	<i>Demo Funktionsweise Software Focus on Bee-Bot durch TR</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind; Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot. TR zeigt dem Kind anhand der Programmierung eines Beispiels die Funktionsweise der Software.				
Positive Notiz:	Keine positiven Auffälligkeiten.				

Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Sollte sich bei den ersten Aufgaben mit der Software herausstellen, dass sich das Kind die Bedienung der Software nicht selbst mit kleinen Hilfestellungen erarbeiten kann, empfiehlt sich eine Demo der Software durch TR, um der Demotivation des Kindes vorzubeugen.

AufgabenID:	401	Schwierigkeitsgrad:	3	Dauer:	5 Minuten
Aktivitätscluster:	E3) Wissenstransfer			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Orientierung in der Software Focus on Bee-Bot</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gebeten: „Bitte kannst du mir beschreiben was Du siehst“. Damit soll erreicht werden, dass sich das Kind mit den verschiedenen Elementen der Oberfläche der Software beschäftigt und eventuell schon überlegt wie die Software funktioniert. Es könnte noch nachgefragt werden: „Hast du so ein ähnliches Programm schon einmal gesehen?“ um die Vorerfahrung des Kindes mit Computerprogrammen generell abzuklären.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Keine Anmerkungen.				

AufgabenID:	402	Schwierigkeitsgrad:	3	Dauer:	2 Minuten
Aktivitätscluster:	E3) Wissenstransfer			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Verknüpfung der Software Focus on Bee-Bot mit schon Bekanntem aus den Übungen mit dem realen Bee-Bot</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gefragt: „Siehst Du einen Zusammenhang zwischen dem, was wir im Programm mit euch bis jetzt gemacht haben und dem, was du hier siehst?“ (hier kann man den Hinweis geben, dass es sich in der Software um den Bee-Bot handelt)				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Hardware Bee-Bot muss schon bekannt sein.				

AufgabenID:	405	Schwierigkeitsgrad:	4	Dauer:	3 Minuten
Aktivitätscluster:	E3) Wissenstransfer			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Erkennen von Unterschieden zwischen dem Bee-Bot in der Software Focus on Bee-Bot und dem Hardware-Bee-Bot</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot , kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gefragt: "Gibt es hier bei den Buttons (Tasten) einen Unterschied zu den Tasten beim echten Bienenroboter (Hardware-Bee-Bot)?"				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Aufgabe zur Steuerung der Software sollte schon gemacht worden sein. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1 – Unterschiede zur Hardware: GO-Befehl ist grüner Pfeil in Symbolleiste, Links-und Rechts-Dreh-Buttons sind nicht gedreht, Clear ist rotes X rechts statt blaues Clear links, Pause-Button (links rotes Pausezeichen statt rechts blauer Pause-Button), Farben der Buttons (rote Richtungs-Buttons statt orange Richtungs-Buttons), kein GO-Button in der Mitte des Navigationsbereichs im Programm, bei Clear wird der Bee-Bot an den Start zurück gestellt, bei Drücken des GO-Buttons wird der Bee-Bot wieder an die Startposition zurückgestellt und fährt von dort die gespeicherten Befehle ab (Kind kann bei der Software nicht wie beim realen Bee-Bot Teilstrecken einspeichern, diese mit GO ausführen, den Speicher leeren u dann die nächste Teilstrecke einspeichern um ans Ziel zu kommen). Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 2 – Unterschiede zur Hardware: es gibt zweites GO in der Symbolleiste, bei Clear wird der Bee-Bot an den Start zurückgestellt, bei Drücken des GO-Buttons wird der Bee-Bot wieder an die Startposition zurückgestellt und fährt von dort die gespeicherten Befehle ab (Kind kann bei der Software nicht wie beim realen Bee-Bot Teilstrecken einspeichern, diese mit GO ausführen, den Speicher leeren und dann die nächste Teilstrecke einspeichern um ans Ziel zu kommen).				

AufgabenID:	413	Schwierigkeitsgrad:	3	Dauer:	3 Minuten
Aktivitätscluster:	E3) Wissenstransfer			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Nimmt das Kind den schwarzen Außenbereich rund um die Matte als Begrenzung wahr?</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gefragt: „Glaubst du kann der Bee-Bot im Programm von der Matte runterfahren?“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Hier geht es darum ob das Kind den Außenbereich rund um die Matte als Pendant zum Außenbereich rund um eine Matte in der Realität des Hardware-Bee-Bots erkennt oder diesen als Begrenzung für den Bee-Bot im Programm sieht.				

AufgabenID:	416	Schwierigkeitsgrad:	4	Dauer:	8 Minuten
Aktivitätscluster:	E3) Wissenstransfer			Sozialform:	Einzel
Aufgabenart:	Erforschen				
Aufgabe:	<i>Auffinden eines Symbols im Programm Focus on Bee-Bot für das Vorgehen „Trockenweiterstellen“ welches als Hilfestellung beim Einspeichern von Wegen mit dem realen Bee-Bot zum Einsatz kommen kann und einprogrammieren einer schwierigen Route.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und soll nun einen noch schwierigeren Weg einspeichern als 3 Schritte nach links / rechts. Als Hilfestellung und Vorbereitung auf diese Aufgabe wird Kind, anknüpfend an die bei der Arbeit mit dem realen Bee-Bot gerne von den Kindern als Hilfestellung beim Einspeichern von Wegen angewandte Technik „Trockenweiterstellen“, gefragt: „Findest du ein Symbol, mit dem man sich die Einzelschritte der Biene anzeigen lassen kann (so wie du das beim Einspeichern von Wegen beim realen Bee-Bot gemacht hast)?“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Voraussetzung: Aufgaben aus der Inhaltsdimension A) Programmierumgebung Hardware und die grundlegende Bedienung der Software zur Bewegung des Bee-Bots sollte bekannt sein.				

AufgabenID:	407	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren				
Aufgabe:	<i>Bee-Bot in der Software Focus on Bee-Bot einen Schritt vorwärts bewegen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird gefragt: „Hast du diese Matte schon einmal gesehen?“ und dann aufgefordert: „Bitte kannst du auf der Matte, auf der der Bee-Bot im Programm steht mit ihm einen Schritt vorwärts gehen?“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Aufgabe zur Steuerung der Software sollte schon gemacht worden sein. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1: Die Startposition des Bee-Bot in der Software Focus on Bee-Bot 1 auf der Startmatte befindet sich links unten mit Blick nach oben – die Perspektive des Kindes, des Bee-Bots im Programm und die Ausrichtung des Navigationsbereichs stimmen überein. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 2: Da die Startposition in der Software Focus on Bee-Bot 2 auf der Startmatte das Feld links oben mit Blick nach unten ist, muss das Kind hier einen Perspektivenwechsel zwischen der Perspektive des Bee-Bots im Programm und der eigenen Perspektive vornehmen, die mit der Ausrichtung des Navigationsbereichs übereinstimmt. Ein Schritt vorwärts für den Bee-Bot sieht aus der Perspektive des Kindes nach einem Schritt nach unten (von den Kindern oft als rückwärts interpretiert) aus.				

AufgabenID:	412	Schwierigkeitsgrad:	1	Dauer:	2 Minuten
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren				
Aufgabe:	<i>Bee-Bot in der Software Focus on Bee-Bot zwei Schritte vorwärts bewegen</i>				

Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot und wird aufgefordert: "Bitte kannst du auf der Matte, auf der der Bee-Bot im Programm steht mit ihm zwei Schritte vorwärts gehen?"
Positive Notiz:	Keine positiven Auffälligkeiten.
Negative Notiz:	Keine negativen Auffälligkeiten.
Allgemeine Erkenntnis:	Aufgaben zur Steuerung der Software und einen Schritt in der Software mit dem Bee-Bot vorwärts zu fahren sollten schon gemacht worden sein. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 2: Da die Startposition in der Software Focus on Bee-Bot 2 auf der Startmatte das Feld links oben mit Blick nach unten ist, muss das Kind hier einen Perspektivenwechsel zwischen der Perspektive des Bee-Bots im Programm und der eigenen Perspektive vornehmen, die mit der Ausrichtung des Navigationsbereichs übereinstimmt. Zwei Schritte vorwärts für den Bee-Bot sehen aus der Perspektive des Kindes nach zwei Schritten nach unten (von den Kindern oft als rückwärts interpretiert) aus. Außerdem muss das Kind erkennen, dass es zwei Schritte vorwärts einspeichern muss, da bei erneutem GO-Drücken (bei einem eingespeicherten Vorwärtsbefehl) das Programm mit dem Fahren des Bee-Bots wieder am Startfeld beginnt und nicht wie beim realen Bee-Bot an dem Feld an dem der Bee-Bot nachdem er beim letzten GO-Drücken alle Befehle abgearbeitet hatte zu stehen kam.

AufgabenID:	414	Schwierigkeitsgrad:	2	Dauer:	3 Minuten
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren				
Aufgabe:	<i>Bee-Bot in der Software Focus on Bee-Bot drei Schritte nach links / rechts bewegen</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind. Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot. Voraussetzung: der Bee-Bot steht auf der Startposition der Startmatte – jetzt wird das Kind aufgefordert: „Bewege den Bee-Bot im Programm drei Schritte nach links / rechts.“				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Voraussetzung: Die grundlegende Bedienung der Software zur Bewegung des Bee-Bots sollte bekannt sein. Das Kind übt hier das Zerlegen eines Weges in Einzelbefehle und besonders das Zerlegen einer Bewegung nach rechts in eine 90 Grad Drehbewegung nach rechts mit anschließenden Vorwärtsbewegungen. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 1: Erschwerend ist für das Kind bei der Software Focus on Bee-Bot 1, dass die Links-und Rechts-Dreh-Pfeile im Programm ohne Drehbewegung gestaltet sind, sodass das Kind rein von den Pfeilen her (ohne auf die Vorerfahrung aus der Arbeit mit dem realen Bee-Bot zurückzugreifen) möglicherweise von Seitwärtsbewegungen des Bee-Bots ausgeht. Allgemeine Erkenntnis am Beispiel der Software Focus on Bee-Bot 2: Erschwerend ist für das Kind bei der Software Focus on Bee-Bot 2, dass es einen Perspektivenwechsel zwischen dem Bee-Bot in der Software und der eigenen Perspektive durchführen muss – soll sich der Bee-Bot nach links bewegen ist dies aus Sicht des Kindes nach rechts.				

AufgabenID:	418	Schwierigkeitsgrad:	4	Dauer:	2 Minuten
Aktivitätscluster:	E4) Selbstgewähltes Problem lösen: Sequenzen			Sozialform:	Einzel
Aufgabenart:	Programmieren				
Aufgabe:	<i>Durch Kind Matte im Mattenauswahlbereich auswählen lassen um auf dieser schwierigere Route zu programmieren.</i>				
Beschreibung (Material und Aufgabensetting):	Benötigt wird: Notebook mit installierter Software Focus on Bee-Bot, kindgerechte Maus, Mauspad, Tisch und zwei Sitzmöglichkeiten, die sowohl für Kind als auch für TR passend sind; Das Kind sitzt neben TR vor einem Notebook / Bildschirm mit dem Startbildschirm der geöffneten Software Focus on Bee-Bot. Das Kind soll Matte im Mattenauswahlbereich auswählen um auf dieser schwierigere Route zu programmieren.				
Positive Notiz:	Keine positiven Auffälligkeiten.				
Negative Notiz:	Keine negativen Auffälligkeiten.				
Allgemeine Erkenntnis:	Hier kann aus einer Vielzahl von Matten im Mattenauswahlbereich ausgewählt werden. Sämtliche Aufgaben aus der Inhaltsdimension A) Programmierumgebung Hardware können als Aufgaben für die Software herangezogen werden.				

A3 Hinweise zur Kodierung Abschluss-Assessment

Das Abschluss-Assessment im Kindergarten ist, ähnlich zu den vorhergegangenen Usability-Evaluationen, mit Fingerspitzengefühl durchzuführen, da die Kinder Prüfungssituationen noch nicht kennen und ihnen auch nicht das Gefühl vermittelt werden sollte, dass sie Aufgaben nicht lösen können. Durch entsprechende Hilfestellungen und spätere Auflösung der Aufgaben soll eine Demotivation der Kinder und das Aufkommen von Desinteresse an der Materie Programmierung bzw. Roboter vermieden werden. Gerade bei den Aufgabenstellungen (Frage 5 und Frage 7 bzw. Frage 8) kann ein Lerneffekt auftreten, der allenfalls bei der Interpretation der Ergebnisse zu berücksichtigen ist.

Da die Kinder im Laufe des Interviews und des Abschluss-Assessments ihre Meinung durch Nachfragen hin und wieder revidieren, gilt die Kodiervorgabe, immer die erste Antwort zu kodieren, um so die unverfälschte Meinung des jeweiligen Kindes zu erfassen.

Zur Kodierung kommen folgende Symbole zum Einsatz: „x“ trifft zu, „-“ trifft nicht zu, „n.g.“ nicht gefragt, „n.v.“ Szene kam nicht vor bzw. hat nicht stattgefunden. Letztere Kodierung kann vor allem bei Fragen mit Abhängigkeiten untereinander auftreten, wo das Kind nur eine der verfügbaren Varianten wählt bzw. beantwortet.

A3.1 AS01 Frage 1: Am besten gefallen

Die erste Kategorie beschäftigt sich mit den Präferenzen der Kinder, was ihnen vom gesamten Programm am besten gefallen hat. Die jeweils zutreffende Antwort ist mit „x“ zu kodieren, die übrigen mit „-“ für nicht zutreffend.

A3.2 AS02 Frage 2: Was ist der Bee-Bot

Die Frage, worum es sich beim Bee-Bot handelt, wird in der zweiten Kategorie behandelt. Wobei eine Kombination AS0201 und AS0202 zu erfassen ist, wenn das Kind mit „Bienenroboter“ o.ä. antwortet. AS0203 ist sinnvoll nur dann möglicherweise als zutreffend zu markieren, wenn AS0201 als nicht zutreffend erfasst wurde. Auch in dieser Kategorie ist die jeweils zutreffende Antwort mit „x“ und die nicht zutreffenden Antworten mit „-“ zu kodieren.

A3.3 AS03 Frage 3: Bee-Bot einschalten

Das Einschalten des Bee-Bots erfolgt durch das Betätigen des Schiebeschalters auf der Unterseite des Bee-Bots. Wenn das Kind den Bee-Bot ohne Nachfragen und Hilfestellungen sofort und selbständig eingeschaltet hat, dann ist AS0301 als zutreffend zu markieren. Sollte das Kind als erstes die GO-Taste betätigt haben, dann ist AS0302 als zutreffend zu kodieren und wenn das Kind erst auf Hilfestellung oder mit weiteren Hinweisen den Bee-Bot einschalten kann, dann ist AS0304 als zutreffend zu erfassen. Sollte das Kind nicht wissen, wie der Bee-Bot einzuschalten ist, dann ist AS0303 und in weiterer Folge gegebenenfalls auch AS0305 entsprechend zu kodieren.

A3.4 AS04 Frage 4: Funktion Bee-Bot

Um zu überprüfen, ob die Kinder die grundlegenden Funktionalitäten des Bee-Bots, im Speziellen die Eingabe von Befehlen und das Losfahren, beherrschen und vielleicht sogar erläutern können, sind die Unterkategorien der Hauptkategorie AS04 zu kodieren. Die ersten beiden Kategorien beschäftigen sich mit der Eingabe von Befehlen und dem Losfahren: Wenn das jeweilige Kind demonstrieren kann, wie der Bee-Bot zu bedienen ist, dann ist AS0401 und wenn das Kind erklären kann, wie der Bee-Bot funktioniert, dann ist AS0402 entsprechend zu markieren. Wenn das Kind auf Nachfrage erklären kann, wie der Bee-Bot bedient werden kann, dann ist AS0403 zu kodieren und wenn das Kind zur Erklärung entsprechende Hilfestellungen braucht, ist das in AS0404 zu vermerken. Sollte es passieren, dass das Kind die Funktionalität des Bee-Bots nicht erklären kann, ist das in AS0406 in

Kombination mit AS0405 zu vermerken, vor allem auch dann, wenn die Testleitung durch zahlreiche Nachfragen und Hilfestellungen die Funktionalität quasi schon erklärt hat.

A3.5 AS05 Frage 5: Zwei Schritte vorwärts

Die erste Praxisaufgabe war, dass die Kinder den Bee-Bot zwei Schritte vorwärts fahren lassen sollten. Wenn ihnen das auf Anhieb geglückt ist, dann ist das in Kategorie AS0501 entsprechend zu erfassen. Die weiteren Kategorien präzisieren die Beobachtungen und Verhaltensweisen der Kinder während der Lösung der Aufgabe. Hierbei ist interessant, ob das Kind die Schritte vorab ausgezählt hat (AS0502), während der Befehlseingabe laut mitgezählt hat (AS0503), Hilfestellungen benötigte (AS0504 und AS0505). Wenn das betreffende Kind, die Aufgabe nicht lösen konnte, ist es interessant, ob es das „nicht schaffen“ schon vorher verbalisierte (AS0506), es dann tatsächlich nicht schaffte und die Aufgabe durch die Testleitung gelöst wurde (AS0507). Da durch das Ausprobieren und Erklären (AS03 bzw. AS04) möglicherweise schon Befehle im Bee-Bot gespeichert sind, ist es von Interesse, ob die Kinder daran dachten, den Speicher zu Beginn der Lösung zu leeren (AS0508), um ungewünschte Befehlsausführungen zu vermeiden.

A3.6 AS06 Frage 6: Bee-Bot losfahren lassen

Die Hauptkategorie AS06 kann zum einen aus den Antworten und Kodierungen der Hauptkategorien AS04 und AS05 abgeleitet werden. Zum anderen fragte die Testleitung die Kinder explizit danach. Die entsprechenden Antworten sind in den beiden Unterkategorien zu kodieren.

A3.7 AS07 Frage 7: Komplexen Weg fahren

Die für die Kinder herausforderndste Fragestellung ist die, einen vorgegebenen komplexeren Weg mit dem Bee-Bot auf der zur Verfügung stehenden Matte zu programmieren und den Bee-Bot diesen Weg dann fahren zu lassen. Hier gibt es zwei Lösungsstrategien:

- a) Das Kind löst die Aufgabe mit sog. Einzelschritten. Dh. der Bee-Bot wird nach jeder Befehlseingabe per Hand so gestellt, wie er nach der Ausführung des jeweiligen Befehls stehen würde. (leichter)
- b) Das Kind löst die Aufgabe im Gesamten, ohne der Hilfestellung des Einsatzes von Einzelschritten. (schwerer)

Wenn Kinder von sich aus die Lösungsstrategie a) wählen, dann sind die Kategorien dieser Hauptkategorie entsprechend zu kodieren. Sollten Kinder von sich aus gleich die Lösungsstrategie b) wählen, dann sind nur die ersten vier Kategorien (AS0701 bis AS0704) zu kodieren und dann die Hauptkategorie AS08.

Kinder, die die komplexe Aufgabenstellung mit Einzelschritten lösten, könnten in einem zweiten Durchgang die Aufgabenstellung auch ohne Einzelschritte lösen. Dann ist auch die Hauptkategorie AS08 entsprechend zu kodieren.

Zu Beginn wird mit den Kategorien AS0701 und AS0702 erfasst, ob das Kind während der Programmierung direkt hinter dem Bee-Bot sitzt. Sollte dies nicht der Fall sein, dann wird es von der Testleitung aufgefordert, sich direkt hinter den Bee-Bot zu setzen. Zum einen fällt es dem Kind so leichter, den Bee-Bot zu programmieren, da das Kind in die gleiche Richtung sieht wie der Bee-Bot selbst. Zum anderen werden die Ergebnisse vergleichbar, da alle Kinder den Bee-Bot aus der gleichen Blickrichtung programmierten.

Ob ein Kind den zu fahrenden Weg vorausplant und sich schon vorab Gedanken über den Weg macht, wird in Kategorie AS0703 abgebildet. Kinder, die während der Programmierung zuerst formulieren, was sie tun möchten und dann anschließend den Befehl eingeben, werden über die Kategorie AS0704 identifiziert.

Mit der Kategorie AS0705 wird die Eigenständigkeit des Kindes bei der Problemlösung erfasst. Dieses Maß wird durch eine vierstufige Skala repräsentiert:

- 0 ... Keine Eigenständigkeit, Kind benötigte sehr viel Hilfe
- 1 ... Wenig Eigenständigkeit, Kind benötigte Hilfe
- 2 ... Kind agierte großteils eigenständig, benötigte wenige Hilfestellungen
- 3 ... Kind agierte eigenständig und benötigte keine Hilfe

Die Anzahl der Versuche bis zu Lösung wird in Kategorie AS0706 erfasst und ob das Kind tatsächlich die Aufgabe löste ist in Kategorie AS0707 zu kodieren. Wurde die Aufgabe durch die Testleitung oder das Kind abgebrochen, so ist das in der Kategorie AS0708 zu vermerken. Wenn das Kind die Aufgabe mit einem der möglichen kürzesten Wege gelöst hat, dann ist das durch Kodierung in der Kategorie AS0709 zu erheben. Die weiteren Kategorien umfassen Problemstellungen, die in der Praxis häufig auftreten: Denkt das Kind von sich aus daran, zu Beginn den Speicher des Bee-Bots zu löschen (AS0710) und denkt das Kind daran, nach dem Löschen den Bee-Bot wieder in seine Ausgangsposition zurück zu stellen (AS0711). Ebenso ist festzuhalten, ob das Kind Probleme mit links / rechts (AS0712) hatte, mit der Art der Drehung des Bee-Bots (AS0713), mit der Bestimmung der Anzahl der Schritte (AS0714) oder mit der Eingabe von Befehlen (AS0714).

A3.8 AS08 Frage 8: Ohne Einzelschritte komplexen Weg

Die Hauptkategorie AS08 wird analog zur Hauptkategorie AS07 kodiert. Zu erfassen ist, ob das Kind von sich aus gleich die Möglichkeit der Programmierung im Gesamten, ohne Einzelschritte wählte (AS0801). Wenn das Kind die Aufgabe mit Einzelschritten löste und der Meinung ist, es kann die Aufgabe auch ohne Einzelschritte lösen, ist das in Kategorie AS08013 zu erfassen. Meinte das Kind, dass die Aufgabenlösung mit Einzelschritten leichter als im Gesamten ist, dann ist das in Kategorie AS0812 zu erfassen.

Wenn bei einem Kind die Aufgabe gar nicht vorkommt, auf Grund von zu vielen Fehlversuchen früher, oder weil die Testleitung meinte, die Aufgabe überfordert das Kind, dann sind die entsprechenden Kategorien mit „n.g.“, nicht gefragt, zu kodieren.

A3.9 AS09 Frage 9: Bee-Bot stoppen

Mit Hilfe dieser Kategorie wird festgestellt, ob das betreffende Kind weiß, dass man einen fahrenden Bee-Bot durch den Druck auf die Taste „GO“ stoppen kann (AS0901). Sollte das Kind die Taste „Clear“ einsetzen, dann ist das in Kategorie AS0902 zu erfassen und wenn das Kind nicht weiß, wie der Bee-Bot zu stoppen ist, dann ist Kategorie AS0903 zu kodieren.

A3.10 AS10 Frage 10: Ausschalten, Einschalten, GO

Wenn das jeweilige Kind die richtige Antwort auf Anhieb gibt, dann ist das in Kategorie AS1001 bzw. für den gegenteiligen Fall in Kategorie AS1002 zu erfassen. Wenn das Kind angibt, es nicht zu wissen, wird das in Kategorie AS1003 kodiert und wenn die Demonstration des gewünschten Verhaltens dazu führt, dass das Kind noch immer nicht glaubt, dass die Befehle nach dem Aus- und Wiedereinschalten gelöscht wurden, ist das durch Kategorie AS1004 zu kodieren. Wenn das Kind mehr Hilfestellungen als das bloße Durchführen der gewünschten Aktionen benötigt, um zu einer Antwort zu kommen, dann repräsentiert das Kategorie AS1005.

A3.11 AS11 Frage 11: Löschen

Ob das Kind weiß, wie am Bee-Bot der Speicher gelöscht wird, wird durch die beiden Unterkategorien repräsentiert.

A3.12 AS12 Frage 12: Anzahl der Befehle, die gespeichert werden können

Die Hauptkategorie AS12 erfasst die Antworten der Kinder auf die Frage, wie viele Befehle der Bee-Bot speichern kann. Hierbei ist darauf zu achten, dass die erste Antwort

entsprechend kodiert wird und allfällige Nachfragen, wenn die erste Antwort nicht korrekt war.

A3.13 AS13 Frage 13: Programmieren erklären

Das Wissen der Kinder zum Thema Programmierung ist in der Kategorie AS13 zu kodieren. Hierbei ist interessant, ob die Kinder als erste Antwort die Verbindung zwischen der Befehlseingabe am Bee-Bot und dem Wort Programmierung herstellen können (AS1301). Auch Speichern beim Bee-Bot kann im weiteren Sinne als Programmierung verstanden werden (AS1303). Wenn Kinder das Einschalten als Programmierung verstehen, dann ist das durch Kategorie AS1304 zu kodieren und wenn Kinder nur die Bewegung des Bee-Bots und nicht die Befehlseingabe als Programmierung verstehen, wird das durch Kategorie AS1305 abgedeckt. Wenn das Kind als erste Reaktion angibt, nicht zu wissen, was Programmieren ist, dann ist Kategorie AS1302 entsprechend zu kodieren.

A3.14 AS14 Frage 14: Präferenzen

Die Hauptkategorie AS14 liefert Erkenntnisse über die Präferenzen der Kinder, mit welchen Tools sie Programmieren lernen wollen bzw. mit welchen Tools andere Kinder Programmieren lernen sollen. Hierzu ist die Unterscheidung zwischen Hardware Bee-Bot (AS1401) und Software (AS1402) interessant, sowie allfällige Gründe, sofern angegeben, wie dem Kind gefällt das eine besser (AS1405) oder weil es ein Spielzeug ist (AS1406). Weiter interessant sind Wahrnehmungen der Kinder, ob etwas spannender bzw. schwieriger werden kann (AS1407) oder dass der Bee-Bot in seiner realen Form eine Vorbereitung auf die Nutzung der entsprechenden Software ist (AS1408).

A3.15 AS15 Frage 15: Programme entwickeln

Ob ein Kind selbst einmal Spiele oder Software, wie die Focus on Bee-Bot Software 1 bzw. 2 programmieren möchte, ist durch die beiden Unterkategorien in der Hauptkategorie AS15 zu kodieren.

A3.16 AS16 Frage 16: Berufswunsch

Ein allfälliger Berufswunsch der Kinder ist durch Kodierung nach Themen, wie Technik (AS1601), Wissenschaft allgemein (AS1602), Soziales bzw. Medizin (AS1603) oder einem ganz anderen Gebiet (AS1604) zu erfassen. Wenn das Kind keinen Berufswunsch angibt (AS1605) ist das ebenso zu kodieren, wie wenn es noch nicht weiß, welchen Beruf es ergreifen möchte (AS1606).

A4 Hinweise zur Kodierung Hardware Usability

Usability-Tests mit Kindern sind mit sehr viel Fingerspitzengefühl durchzuführen, da Misserfolge bei der Bedienung (egal ob durch das Kind oder die Hard- oder Software verursacht) sehr schnell zu einer Demotivation der Kinder führen können. Es gilt daher Kindern unmittelbar nach der Bearbeitung einer Aufgabe eine allfällige richtige Antwort bzw. Lösung zu zeigen und zu erklären. Somit entsteht über die Dauer des Usability-Tests natürlich ein Lerneffekt, der in der Beobachtung und der Beurteilung des Usability-Tests berücksichtigt werden muss.

Es ist in Folge bei der Kodierung das jeweils erste Auftreten einer Kategorie in der Beobachtung zu berücksichtigen, da dies die erste Antwort bzw. Intuition des Kindes im Test widerspiegelt. Das mehrfache Auftreten eines Beobachtungsaspekts ist bei der Kodierung außen vor zu lassen, da im Rahmen des Tests möglichst alle Funktionen erkundet werden sollten und nicht die Kinder trainiert werden, selbständig Aufgaben mit dem Bee-Bot zu lösen. Das mehrfache Auftreten eines Aspekts kann zum einen auf eine tatsächliche Usability-Schwäche hindeuten oder zum anderen auch weitere Effekte, die der Situation, in der sich das Kind zum Zeitpunkt des Tests befindet, geschuldet sind (z.B. Konzentration um Neues zu erlernen und zu merken, Aufmerksamkeit, ...), aufzeigen. Im Fokus des Interesses bei der Kodierung steht der Bee-Bot und seine Usability, jedoch nicht die Kinder selbst.

Zur Kodierung kommen folgende Symbole zum Einsatz „x“ trifft zu, „-“ trifft nicht zu, „n.g.“ nicht gefragt, „n.v.“ Szene kam nicht vor bzw. hat nicht stattgefunden.

Um die Bedienung von Tasten am Bee-Bot zu erfassen, dient folgende Kodierungstabelle:

Taste am Bee-Bot	Code
Vorwärts	VO
Rechts	RE
Rückwärts	RW
Links	LI
Go	GO
Pause	PA
Clear	CL
Einschalten	ON
Ausschalten	OF
Ton einschalten	SO
Tonausschalten	SF

Tabelle 27: Kodierung Tasten am Bee-Bot

A4.1 HW01 Physikalische Eigenschaften des Bee-Bots

Die Hauptkategorie „Physikalische Eigenschaften des Bee-Bots“ beschäftigt sich mit der Nennung wesentlicher physikalischer Merkmale am Bee-Bot. Dazu ist die jeweilige Kategorie als erfüllt, mit „x“, zu kodieren, wenn das Kind diese das erste Mal erwähnt. Mögliche Nachfragen seitens der Testleitung und das in weiterer Folge Hinführen des Kindes auf die richtige Antwort, was vor allem aus motivationalen Aspekten geschehen ist, bleiben außen vor und sind nicht zu berücksichtigen.

A4.2 HW02 Aktionen des Bee-Bots

Die Hauptkategorie „Aktionen des Bee-Bots“ erfasst die vom Bee-Bot ausführbaren Aktionen. In Summe existieren drei Aktionen, die der Bee-Bot für seine Umwelt wahrnehmbar durchführen kann: fahren, piepen und mit den Augen leuchten. Da kein einziges Kind im Hardwaretest das Piepen des Bee-Bots bei einem Tastendruck oder bei der Abarbeitung der Befehle erwähnte, scheint diese Art von Feedback für Kinder uninteressant, nicht wahrnehmbar oder nicht erwähnenswert. Folglich kam es zu keiner Ableitung einer entsprechenden Kategorie. Das visuelle Feedback des Bee-Bots, welches simultan mit dem

Piepen einhergeht, fiel mehreren Kindern auf und wurde auch von ihnen benannt. Somit ist die jeweils erste Nennung dieser Aktion zu kodieren. Die Aktion „Fahren“ des Bee-Bots steht in weiterer Folge im Zentrum des Interesses und wird daher in anderen Kategorien behandelt.

A4.3 HW03 Bedienung des Bee-Bots

Die „Bedienung des Bee-Bots“ ist die Kernfrage des Hardware Usability Tests. Welche Taste drückt das jeweilige Kind als erstes (HW0301, auf die Frage wie der Bee-Bot funktioniert) ist hier ebenso interessant, wie ob ein Kind den Bee-Bot selbständig oder mit Hilfe und Hinweisen einschalten kann (HW0302 und HW0303). Weiter interessant ist die Tatsache, ob Kinder den Bee-Bot manuell schieben möchten (HW0304), ob sie intuitiv die Funktionalität der Tasten erläutern können (HW0305, HW0306) und diese dann auch sinngemäß nutzen können (HW0309, HW0310). Hierzu sind zusätzlich die Kategorien „HW0307 bringt Bee-Bot zum Fahren“ und „HW0314 eine geeignete Befehls-Reihenfolge“ zu kodieren. Mit diesen Kategorien kann unterschieden werden, ob die Tatsache, dass der Bee-Bot eingeschaltet ist oder nicht, eine Rolle für die Kinder spielt oder nicht bzw. ob Kinder den Bee-Bot im ausgeschalteten Zustand intuitiv korrekt bedienen würden.

A4.4 HW04 Herangehensweisen und Eigenschaften des Kindes

Das Verhalten der Kinder während des Usability-Tests des Bee-Bots wird vor allem durch ihre Herangehensweisen und ihre Interaktionen erfasst. Bei diesen Kategorien handelt es sich jeweils um einen Gesamteindruck über den gesamten Test hinweg. Wenn ein Kind sich traut den Bee-Bot anzugreifen und nicht lange zögert ihn das erste Mal zu berühren, dann ist HW0401 mit einem „x“ zu kodieren. Wenn ein Kind eine Äußerung tätigt, die auf Angst vor dem Bee-Bot schließen lässt, dann ist HW0402 mit einem „x“ zu kodieren. Wenn ein Kind mit dem Bee-Bot spricht, dann ist HW0403 mit einem „x“ zu kodieren. Ein Kind gilt als experimentierfreudig (HW0404), wenn es keine Berührungängste zeigt, häufig die Tasten des Bee-Bots nutzt, ohne Angst zu zeigen, etwas falsch zu machen und wenn ein Kind auf mögliche Fragestellungen mit Ausprobieren am Bee-Bot reagiert. Als zurückhaltend (HW0405) ist zu kodieren, wer wenig mit der Testleitung spricht, sich nicht traut den Bee-Bot auszuprobieren oder wer oftmals ermutigt werden muss, eine Taste zu drücken. Mit HW0406 ist festzustellen, ob ein Kind dem fahrenden Bee-Bot ausweicht oder nicht. Die Interaktion mit der Testleitung (HW0407) ist mit „x“ zu kodieren, wenn das Kind in das von der TL geführte Gespräch mit einsteigt, aktiv antwortet, eventuell sogar eigene Fragen stellt und ein Gesprächsfluss zwischen TL und dem Kind vorhanden ist. Wenn sich die Interaktion mit der TL seitens des Kindes im Verlauf des Tests steigert, dann ist HW0408 mit „x“ zu kodieren.

A4.5 HW05 Interesse an Bee-Bot

Da kein Kind kein Interesse am Bee-Bot gezeigt hat, wurde in weiterer Folge unterschieden, ob generelles Interesse am Bee-Bot besteht oder ein sehr starkes Interesse am Bee-Bot besteht. (HW05) Ob starkes Interesse vom Kind gezeigt wird, ist primär durch das Verhalten des Kindes zu beurteilen, vor allem aber auch durch die Experimentierfreude, keine Berührungängste, den Grad der Interaktion während des Tests und ob das Kind von sich aus ebenso Fragen stellt.

A4.6 HW06 Analogien zur Software Focus on Bee-Bot 1 bzw. 2

Die Hauptkategorie HW06 „Analogien zur Software Focus on Bee-Bot 1 bzw. 2“ ist speziell für den HWT2 im Kindergarten E relevant. Bei diesem Testsetting konnten die Kinder die Software Focus on Bee-Bot 1 bzw. 2 vor dem Hardware Bee-Bot kennenlernen. Um herauszufinden, ob die Kenntnis der Software einen Vorteil in der Bedienung der Hardware mit sich bringt, sind die entsprechenden Unterkategorien zu kodieren. HW0601 und HW0602 beschäftigen sich mit dem Wiedererkennen des Bee-Bots, der Bedienelemente und der Bedienung des Bee-Bots. Die Testleitung hat die Funktionalität Clear bereits bei der Vorstellung der Software besprochen, daher ist hier von Interesse, ob dieses Wissen auf die

Bedienung der Hardware transferiert werden kann (HW0603 und HW0604). Ob den Kindern generell Unterschiede zwischen der Hardware und Software auffallen (in punkto Bedienung, Reaktionen, Aktionen setzen etc.) ist mit der Kategorie HW0605 zu kodieren und ob diese Unterschiede zu (anfänglichen) Irritationen führen ist in Kategorie HW0606 zu kodieren. Abschließend stellt die Testleitung den Kindern die Frage, ob sie zum Programmieren die Hardware oder die Software bevorzugen (HW0607 und HW0608).

A5 Hinweise zur Kodierung Software Usability

Da die Software Usability Tests mit viel Fingerspitzengefühl, wie bereits in Kapitel 9.2.2.4 dargelegt, durchzuführen sind, entsteht über die Dauer des Software Usability Tests natürlich ein Lerneffekt, der in der Beobachtung und der Beurteilung des Software Usability Tests berücksichtigt werden muss.

Es ist in Folge bei der Kodierung das jeweils erste Auftreten einer Kategorie im Transkript des Software Usability Tests zu berücksichtigen, da dies die erste Antwort bzw. Intuition des Kindes im Test widerspiegelt.

Das mehrfache Auftreten eines Aspekts ist bei der Kodierung außen vor zu lassen, da im Rahmen des Software Usability Tests möglichst alle Funktionen erkundet werden sollten und nicht die Kinder trainiert werden, selbständig Aufgaben mit der Software Focus on Bee-Bot 1 bzw. 2 zu lösen. Von daher kann das mehrfache Auftreten eines Aspekts zum einen auf eine tatsächliche Usability-Schwäche hindeuten oder zum anderen auch weitere Effekte, die der Situation, in der sich das Kind zum Zeitpunkt des Tests befindet, geschuldet sind (z.B. Konzentration um Neues zu erlernen und zu merken, Aufmerksamkeit, ...), aufzeigen. Im Fokus des Interesses bei der Kodierung stehen die Software Focus on Bee-Bot 1 bzw. 2 und ihre Usability, jedoch nicht die Kinder selbst.

Die Kodierung erfolgt analog zu den bisherigen Kodierungen: „x“ wird für trifft zu kodiert, „-“ für trifft nicht zu, „n.g.“ für nicht gefragt, „n.v.“ Szene kam nicht vor bzw. hat nicht stattgefunden, Kind hat nicht geantwortet oder die Testleitung hat die Aufgabenstellung für das Kind gelöst. Bei jeder Kodierung ist eine fortlaufende Nummer anzuführen, die im Transkript ebenfalls an der Stelle angeführt wird, aus der sich die Kodierung der jeweiligen Kategorie ableiten lässt, um die Kodierung nachvollziehbar zu gestalten.

Der Begriff „interpretiert“ umfasst bei der Kodierung alle Tätigkeiten eines Kindes, die dazu geeignet sind, dass eine Beobachterin oder ein Beobachter in der Lage ist zu beurteilen, ob das Kind die jeweilige Lösung bzw. Antwort parat hat. Zu diesen Ableitungen können eine Aussage des Kindes, eine Handlung oder eine Geste dienen.

Kategorien, die zusammenhängen und aus denen immer nur eine Kategorie mit „x“ kodiert werden kann, Single Choice, sind als Gruppe auf der rechten Seite mit einem Doppelbalken gekennzeichnet.

A5.1 SW01 Vorwissen

Die Hauptkategorie „Vorwissen“ dient zur Erfassung des Vorwissens der einzelnen Kinder, um die Ergebnisse in weiterer Folge entsprechend interpretieren zu können. Wenn ein Kind die entsprechenden Fragen beantwortet hat, dann ist die jeweilige Kategorie mit „x“ zu kodieren.

A5.2 SW02 Wahrnehmung

Mit Hilfe der Hauptkategorie „Wahrnehmung“ soll festgestellt werden, was die Kinder von der Bildschirmoberfläche wahrnehmen und wie sie die Grundfunktionalitäten zu Beginn einschätzen. Wenn das Kind im Bee-Bot eine Biene erkennt, ist das in der Kategorie SW0201 entsprechend zu kodieren. Handelt es sich laut Aussage des Kindes um einen Roboter, dann ist Kategorie SW0202 mit „x“ zu kodieren, wenn es sich um keinen Roboter handelt, weil es ein Tier ist, dann Kategorie SW0203. Erkennt ein Kind im Bee-Bot ein anderes Tier, dann ist das in der Kategorie SW0204 entsprechend zu kodieren. In weiterer Folge soll erfasst werden, wie die Kinder die Funktionalität des Bee-Bots interpretieren: Kann der Bee-Bot fahren (SW0205), springen (SW0206), fliegen (SW0207) oder über Hürden springen, fliegen o.ä. (SW0208). Ob den Kindern der Befehlsbereich (standardmäßig links unten am Bildschirm) auffällt, ist mit der Kategorie SW0209 zu erfassen.

Die Testleitung hatte den Kindern im Vergleich zum realen Bee-Bot den realen Constructa-Bot gezeigt (die Funktion desselben deckt sich mit der des Bee-Bots, nur das Aussehen differiert). Glauben die Kinder, dass es sich hierbei um einen Roboter handelt (SW0210) oder erkennen sie im Constructa-Bot ein Baugerät (SW0211), so ist das in den entsprechenden Kategorien zu kodieren. Kategorie SW0212 dient dazu zu dokumentieren, ob Kinder zu Beginn des Testsettings nur Details von oder auf der Matte erwähnen.

A5.3 SW03 Programmlogik / Benutzeroberfläche: Bedienung Software

Die größte Hauptkategorie, SW03 „Programmlogik / Benutzeroberfläche: Bedienung Software“, beinhaltet alle relevanten Bedien- und Steuerelemente der Software-Oberfläche.

Zu Beginn ist zu erfassen, ob das Kind das Programm mit der Maus (SW0301), mit einem anderen Eingabegerät (SW0302) oder mit Spracheingabe (SW0303) steuern möchte.

Für beide Software-Versionen separat (V1 für Focus on Bee-Bot 1 und V2 für Focus on Bee-Bot 2) ist zu kodieren, ob das jeweilige Kind keine Probleme hat zwischen den Clear- und Pause-Buttons zu unterscheiden (SW0304) und ob das jeweilige Kind den GO-Button als Vorwärts-Befehl interpretiert (SW0307). Weiter ist für beide Software-Versionen getrennt zu kodieren, ob das jeweilige Kind interpretiert, dass der Bee-Bot durch Klicken auf einen Befehl in der Befehlsliste gesteuert werden kann (SW0319) oder eine beliebige Position auf der Oberfläche (außer den GO-Button) für den GO-Button hält (SW0321).

Nur für die Software-Version V1 ist zu kodieren, ob das jeweilige Kind interpretiert, dass der Bee-Bot im Navigationsbereich (links oben am Bildschirm) auf dem Rücken bedient werden kann (SW0311). Nur für die Software-Version V2 ist zu erfassen, ob die Kinder den GO-Button im Navigationsbereich als „Start“ für das Losfahren des Bee-Bots interpretieren (SW0306).

Die folgenden Kategorien sind unabhängig von der jeweiligen Software-Version zu kodieren:

- Interpretiert das Kind die Links- und Rechts-Dreh-Buttons als 90 Grad Drehung und nicht als Schritt nach links oder rechts (SW0305)?
- Wird von dem Kind interpretiert, dass der Bee-Bot auf der Matte auf seinem Rücken bedient / gesteuert wird (SW0308)?
- Erkennt das Kind den Navigationsbereich ohne Hilfestellung (SW0309) oder mit Hilfestellung (SW0310)?
- Interpretiert das Kind, dass der Bee-Bot durch Klick auf die Matte (oder ein Symbol auf der Matte) in seiner Richtung gesteuert werden kann (SW0312)?
- Findet das Kind selbst den GO-Button bzw. den grünen Pfeil in der Symbolleiste als Befehl für das Losfahren des Bee-Bots (SW0313)?
- Findet das Kind mit max. einem Hinweis den Clear-Button zum Löschen auf Anhieb (SW0314) oder interpretiert das Kind einen beliebigen Button (nicht den Clear-Button) als Clear-Button (SW0315)?
- Findet das Kind den Button für die Funktionalität „Einzelschritte anzeigen“ auf Anhieb und mit max. einem Hinweis (SW0316) oder interpretiert das Kind einen beliebigen Button in dieser Richtung (SW0317)?
- Interpretiert das jeweilige Kind die Benutzeroberfläche so, dass der Bee-Bot durch beliebige Mausinteraktionen (wie Drag-&-Drop oder das Schieben des Bee-Bots) gesteuert werden kann (SW0318)?
- Benötigt das Kind Hilfestellung zur Eingabe einer geeigneten vollständigen Befehlsabfolge (zur Lösung einer Aufgabe) (SW0320)?
- Finden die Kinder es gut oder sind nicht irritiert, wenn der Bee-Bot immer von seinem Startfeld aus beginnt alle eingespeicherten Befehle abzuarbeiten (SW0322)?

- Kann das jeweilige Kind die auftretenden Pop-Ups lesen, sinngemäß interpretieren und entsprechend handeln (SW0323)?
- Thematisiert das Kind von sich aus, dass der Bee-Bot nach Drücken eines beliebigen Richtungs-Buttons nicht losfährt (SW0324)?

A5.4 SW04 Sinngemäße Bedienung

Wie es um eine sinngemäße Bedienung der Software Focus on Bee-Bot 1 bzw. 2 durch die Kinder bestellt ist, ist durch die Hauptkategorie SW04 „Sinngemäße Bedienung“ zu erfassen.

Wenn die Kinder die Funktionalität von GO sinngemäß interpretieren können, ist das entsprechend in der Kategorie SW0401 zu kodieren. Ebenso ist festzuhalten, ob die Kinder bei der Focus on Bee-Bot 2 Software häufiger den GO-Button im Navigationsbereich als den GO-Button in der Symbolleiste nutzen (SW0402).

Ist das Kind in der Lage eine für die Aufgabenstellung geeignete Abfolge von Richtungsbefehlen ohne Hilfestellung einzugeben (SW0403) und bringt es anschließend meistens den Bee-Bot ohne Hilfestellung zum Fahren (SW0404) ist das in den jeweiligen Kategorien zu kodieren. Wenn das Kind erkennt, dass der von ihm eingespeicherte Weg nicht der kürzeste bzw. schnellste war, ist das in Kategorie SW0405 zu vermerken.

Auf den Matten, auf denen sich der Bee-Bot in der Software bewegt, sind zum Teil Hürden vorhanden. Wenn Kinder diese Hürden als unüberwindbares Hindernis erkennen (SW0406), zusätzlich erkennen, dass das Hindernis umfahren werden muss (SW0407) und sie es dann tatsächlich umfahren (SW0408) ist das in den entsprechenden Kategorien abzubilden.

Ob das Kind versteht, was die Funktion „Löschen“ im Kontext des Bee-Bots und der Software Focus on Bee-Bot 1 bzw. 2 bedeutet, ist in der Kategorie SW0409 zu kodieren. Wenn den Kindern auffällt, dass es aktive und nicht-aktive Buttons auf der Benutzeroberfläche gibt und sie die Unterschiede thematisieren bzw. darauf aufmerksam machen, ist das in der Kategorie SW0410 festzuhalten. Wenn das Kind der Meinung ist, dass der Bee-Bot „vergessen“ (Anm. d. Verf.: als Synonym für Löschen in den Einheiten verwendet worden) kann, dann ist das in der Kategorie SW0411 entsprechend zu kodieren.

In der Kategorie SW0412 ist zu dokumentieren, ob das Kind thematisiert, dass Befehle bei ihrer Eingabe aneinandergereiht werden. Die Kategorie SW0413 bildet ab, ob ein Kind die Funktionalität „Einzelschritte anzeigen“ durch das Betätigen des GO-Buttons nach (fast) jeder Betätigung eines Richtungs-Buttons simuliert. Wenn das Kind vermutet, dass es den GO-Button sooft betätigen muss, wie Richtungs-Buttons gedrückt wurden, dann ist Kategorie SW0414 entsprechen zu kodieren. Wenn Kinder die Funktion „Einzelschritte anzeigen“ aktiviert haben und der Meinung sind, dass dann, nach der vollständigen Richtungs-Befehls-Eingabe, das Betätigen des GO-Buttons nichts bewirkt, ist dies in Kategorie SW0415 zu dokumentieren. Wenn es scheint, dass das Kind nach Erklärung die Funktionalität „Einzelschritte anzeigen“ versteht (SW0416) ist das ebenfalls zu erfassen, wie wenn das Kind durch den Wechsel zwischen den Modi „Einzelschritte anzeigen an“ und „Einzelschritte anzeigen aus“ irritiert ist (SW0417).

A5.5 SW05 Orientierung

Das Themengebiet der Orientierung beschäftigt sich vor allem mit den unterschiedlichen Perspektiven, die sich für den User aufgrund der Positionierung des Bee-Bots auf der Matte und der Ausrichtung derselben ergeben. In der Version 1 der Software Focus on Bee-Bot steht der Bee-Bot bei Programmstart am linken unteren Feld der Matte mit Blickrichtung nach oben, in Blickrichtung der Richtungs-Buttons im Navigationsbereich (und des Users), so dass ein Klick auf Vorwärts auch ein Vorwärtsbewegen des Bee-Bots aus der Perspektive des Users bewirkt. In der Version 2 steht der Bee-Bot entgegengesetzt der Blickrichtung der Richtungs-Buttons im Navigationsbereich (und des Users), so dass ein Klick auf Vorwärts zwar Vorwärts für den Bee-Bot bedeutet, dieser sich aber vom oberen Ende der Matte zum

unteren Ende der Matte bewegt – sich also aus der Perspektive des Users nach unten bewegt.

Die Abbildung 40 auf Seite A-48 zeigt ein weiteres Beispiel für unterschiedliche Perspektiven in der Software Focus on Bee-Bot 2. Ein Kind möchte, dass der Bee-Bot (Startposition des Bee-Bots im linken oberen Feld der Matte) einen Schritt vorwärts gehen soll. Es betätigt im Navigationsbereich die Rückwärts-Taste, da diese aus der Perspektive des Kindes in die Richtung zeigt, in der das Feld liegt, auf das der Bee-Bot fahren soll. Tatsächlich führt der Bee-Bot einen Schritt rückwärts aus und fährt von der Matte. Das Kind versucht erneut den Bee-Bot in seine Richtung zu bewegen. Die Art der Darstellung der eingegebenen Befehle im Befehlsbereich links unten am Bildschirm unterstützt die Irritation des Kindes, welches nicht lesen kann und sich daher an der Pfeilrichtung orientiert.

Wenn das jeweilige Kind in der Lage ist, durch Rotation der Matte, die unterschiedlichen Perspektiven des Bee-Bots auszugleichen und sich dadurch behilft, die Aufgaben zu lösen, so ist das entsprechend in der Kategorie SW0501 zu kodieren. In der Kategorie SW0502 soll erfasst werden, ob das Kind in der Lage ist, die Richtungs-Buttons, trotz einer anderen Blickrichtung des Bee-Bots in Relation zum Navigationsbereich (mit den Richtungs-Buttons), korrekt zu bedienen.

Wenn der Bee-Bot in Relation zum Navigationsfeld in die gleiche Richtung sieht, könnte dem jeweiligen Kind die Bedienung des Programms bzw. des Bee-Bots leichter fallen und dies ist in Kategorie SW0503 zu erfassen.

Sollte das jeweilige Kind Probleme mit der Unterscheidung / Umsetzung von Links- und Rechts-Dreh-Befehlen haben (SW0504) oder Richtungsfehler selbst erkennen und korrigieren können (SW0505), ist das in den entsprechenden Kategorien zu erfassen.



Abbildung 40: Perspektiven in Software Focus on Bee-Bot 2

A5.6 SW06 Haptik

Mit der Hauptkategorie SW06 „Haptik“ wird das Bedienen der Computermaus durch das Kind erfasst. Sollte das Kind nach Abrutschen der Maus den Cursor nicht mehr am Bildschirm finden (SW0601), oder bemerkt das Kind nicht, dass es bei einem Mausklick auf einen Button abgerutscht ist und in weiterer Folge der Befehl nicht gespeichert bzw. ausgeführt wurde (SW0602), dann ist dies in den entsprechenden Kategorien festzuhalten.

Rückt das Kind näher zum Bildschirm, um den Inhalt besser erkennen zu können (SW0603) oder hat es Probleme bei der Nutzung der Computermaus (Positionierung, Tasten erreichen, klicken, ...) (SW0604), so ist das in den jeweiligen Kategorien entsprechend zu kodieren. Ebenso festzuhalten ist, ob das Kind mit der Computermaus ohne Umwege zum gewünschten Zielobjekt navigieren kann (SW0605).

A5.7 SW07 Allgemeine Fehler

Sollte das Kind unabsichtlich die Matte drehen (SW0701), die Matte nicht in die Ausgangsposition drehen können (SW0702) oder das Drehen der Matte selbständig beenden können (SW0704) ist das in den jeweiligen Kategorien der Hauptkategorie SW07 „Allgemeine Fehler“ zu kodieren. Ebenso ist in dieser Hauptkategorie zu kodieren, wenn das Kind nicht erkennt, dass durch das Programm unausführbare Befehle (z.B. wegen Hürden) automatisch gelöscht werden (SW0703).

A5.8 SW08 Unterschiede

Die Unterschiede zwischen den Programmversionen werden in der Hauptkategorie SW08 erfasst und kodiert.

- Erkennt das Kind, dass in der Focus on Bee-Bot 2 Software zwei Möglichkeiten für GO zur Verfügung stehen (SW0801)?
- Findet das Kind die Matten der Version 1 oder Version 2 schöner / besser (SW0802)?
- Welche Version empfindet das jeweilige Kind als leichter (zu bedienen) (SW0803)?
- Welcher Navigationsbereich wird vom jeweiligen Kind bevorzugt, da das Kind die Bedienung als leichter empfindet (SW0804)?
- Thematisiert das Kind von sich aus den Unterschied in den Links- und Rechts-Dreh-Buttons der beiden Software-Versionen (SW0805)?
- Welche Software-Version gefällt dem jeweiligen Kind ohne Angabe weiterer Gründe besser (SW0806)?
- Bei welcher Software-Version finden die Kinder die Steuerung bzw. Bedienung besser (SW0807)?
- Welche Software-Version kommt beim jeweiligen Kind auf Grund der optischen Gestaltung besser an (SW0808)?

A5.9 SW09 Wissenstransfer

Die Kodierung der Hauptkategorie SW09 „Wissenstransfer“ dient der Untersuchung eines möglichen Wissenstransfers vom Hardware Bee-Bot auf die Software Focus on Bee-Bot 1 bzw. 2 oder zwischen den beiden Software-Versionen.

Die Kodierung eines möglichen Wissenstransfers zwischen dem realen Bee-Bot und der Software Focus on Bee-Bot 1 bzw. 2 kann nur für die Kinder aus der Gruppe SWT1 erfolgen, da nur diese den realen Bee-Bot bereits im Vorfeld kennengelernt haben. Dazu zählen die folgenden Kategorien:

- Erkennt das Kind, dass die Matten in der Software grundsätzlich so aussehen, wie die in Natura zur Verfügung stehenden (SW0901)?
- Thematisiert das Kind Unterschiede zum realen Bee-Bot (SW0902)?

- Als Ausnahme kann für beide Gruppen (SWT1 und SWT2) die Kategorie SW0903 kodiert werden, die erfasst, ob Kinder meinen, dass der reale Bee-Bot stechen könnte oder böse sei.
- Erkennt das Kind den realen Bee-Bot in der Software als Bee-Bot wieder (SW0904)?
- Erkennt das Kind in der jeweiligen Programmversion (V1 oder V2) die Unterschiede bei den Buttons zwischen dem realen Bee-Bot und den Buttons im Navigationsfeld der Software (SW0905)?
- Kann das Kind seine Erfahrung und sein Wissen aus der Nutzung der Hardware in die Nutzung der Software übertragen (SW0910)?
- Kann das Kind seine Erfahrung und sein Wissen aus der Nutzung der Hardware und der Software Focus on Bee-Bot 1 bzw. 2 auf die Nutzung des Constructa-Bots übertragen (SW0911)?
- Thematisiert das Kind von sich aus das Zurückstellen des Bee-Bots auf sein Startfeld in der Software, sobald der GO-Button gedrückt wird (SW0912)? Dies tritt dann auf, wenn der Bee-Bot beim Drücken des GO-Buttons nicht auf seiner Startposition steht, im Unterschied zum realen Bee-Bot, der seine Position nie selbständig verändert.
- Erkennt das Kind bei der Software Focus on Bee-Bot, dass Buttons ausgegraut sein können (im Moment nicht verfügbar) als Unterschied zum realen Bee-Bot, bei dem stets alle Tasten verfügbar (also drückbar) sind (SW0913)?

Einzelne Kategorien, wie SW0906, SW0907, SW0908 oder SW0909 behandeln die Unterschiede zwischen den beiden Programmversionen V1 und V2 und können daher für beide Gruppen, SWT1 und SWT2, kodiert werden:

- Erkennen die Kinder die Unterschiede zwischen den Buttons zwischen den beiden Software-Versionen (SW0906)?
- Gibt ein Kind an, dass es durch die aufgefundenen Unterschiede irritiert ist (SW0907)?
- Kann das Kind seine Erfahrung bzw. sein Wissen aus der Nutzung der Software Focus on Bee-Bot 1 auf die Nutzung der Software Focus on Bee-Bot 2 übertragen (SW0908)?
- Fällt es dem Kind schwer, die Lösung einer Aufgabe aus der realen Welt in die virtuelle Welt zu übertragen (SW0910)?

A5.10 SW10 Präferenzen

Die Hauptkategorie SW10 dient dazu, die Präferenzen der Kinder für einzelne Funktionen und Aktivitäten zu erfassen.

Wenn ein Kind die Funktion „Einzelschritte anzeigen“ als hilfreich empfindet, ist das entsprechend in der Kategorie SW1001 zu verzeichnen. Finden Kinder den Bee-Bot ansprechender als den Constructa-Bot, so ist das in der Kategorie SW1002 zu kodieren. In der Kategorie SW1003 soll ersichtlich werden, ob einem Kind der Constructa-Bot besser als der Bee-Bot gefällt (SW1003). Um „besser gefällt“ beurteilen zu können, zählen Begründungen wie „lustiger“, „kann was transportieren“, „kann zeichnen“ oder „da passt ein Stift rein“ zu dieser Kategorie dazu. Mit der Kategorie SW1004 wird erfasst, ob die Kinder die Software gegenüber dem realen Bee-Bot bevorzugen.

A5.11 SW11 Testsetting

Um das Testsetting allgemein evaluieren zu können, sind die Kategorien der Hauptkategorie SW11 „Testsetting“ zu kodieren.

Stellt sich ein Kind von selbst (weitere) Aufgabenstellungen mit dem Bee-Bot, so ist das durch die Kategorie SW1101 abzubilden. Kommentiert das Kind sein eigenes Handeln (SW1102), stellt es von sich aus Fragen an die Testleitung (SW1103) oder interagiert mit der

Testleitung (antwortet das Kind auf Nachfragen) (SW1104), dann ist das in den entsprechenden Kategorien zu kodieren. Ignoriert das Kind die Aufgabenstellung und klickt auf beliebige Stellen auf der Software-Oberfläche, ist das durch die Kategorie SW1105 zu erfassen. Ob sich ein Kind durch andere Programmelemente während einer Aufgabe ablenken lässt (SW1106) oder meint, dass es die Aufgabe nicht lösen kann (SW1107) soll die Kodierung der entsprechenden Kategorien zeigen.

A6 Symbole

A6.1 Symbole Gruppe 1

a) Aufgabenauswahl		
		
b) Programm beenden		
		
c) Hilfe		
		
		
d) Einen Schritt vorwärts gehen		
		
Symbole d und e	Symbole d, e, f und g	
e) Einen Schritt rückwärts gehen		
f) Nach links drehen		
		
Symbole f und g	Symbole f und g, oder d und e	
g) Nach rechts drehen		

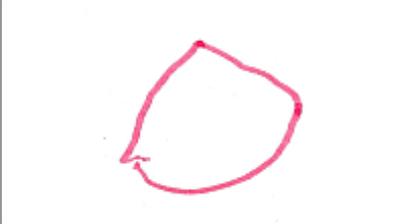
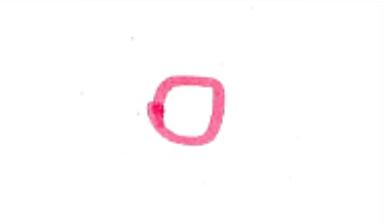
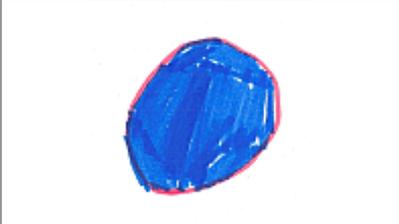
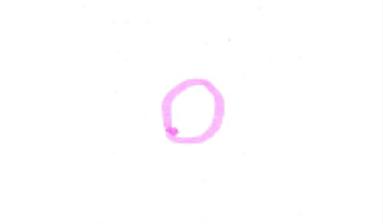
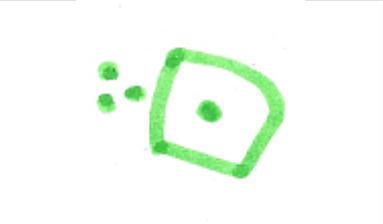
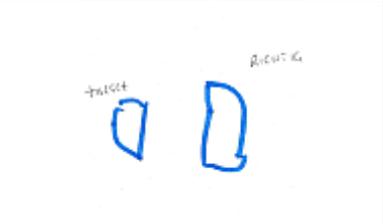
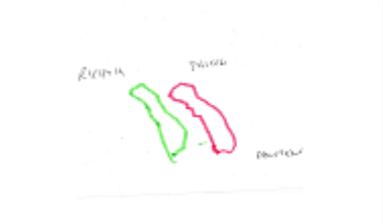
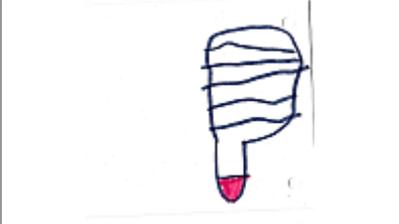
h) Start		
		
i) Stopp		
		
j) Löschen		
		
k) Pause		
		
l) Aufgabe geschafft		
		
m) Aufgabe nicht geschafft		
		

Tabelle 28: Symbole der Kinder aus Gruppe 1

A6.2 Symbole Gruppe 2

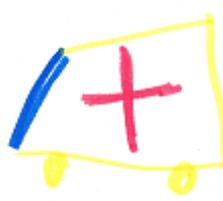
a) Aufgabenauswahl		
b) Programm beenden		

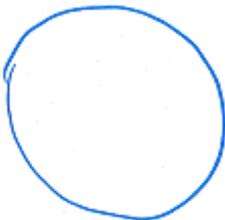
c) Hilfe		
		
d) Einen Schritt vorwärts gehen		
		
Symbole d, e, f und g	Symbole d, e, f und g	Symbole d, e, f und g
e) Einen Schritt rückwärts gehen		
f) Nach links drehen		
g) Nach rechts drehen		
h) Start		
		
i) Stopp		
		
j) Löschen		
		
k) Pause		
		

l) Aufgabe geschafft		
		
m) Aufgabe nicht geschafft		
		

Tabelle 29: Symbole der Kinder aus Gruppe 2

A6.3 Symbole Gruppe 3

a) Aufgabenauswahl		
		
b) Programm beenden		
c) Hilfe		
		
d) Einen Schritt vorwärts gehen		
		
Symbole für d, e, f und g		
e) Einen Schritt rückwärts gehen		
		

f) Nach links drehen		
		
g) Nach rechts drehen		
		
h) Start		
		
i) Stopp		
		
j) Löschen		
		
k) Pause		
		
l) Aufgabe geschafft		
		

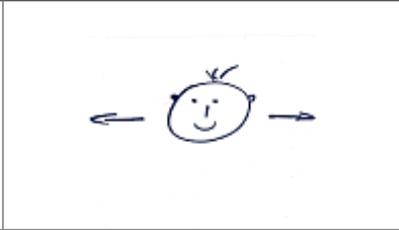
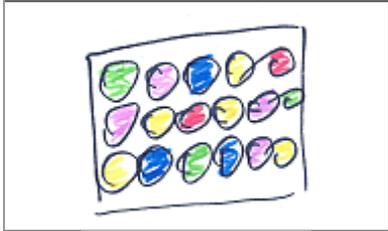
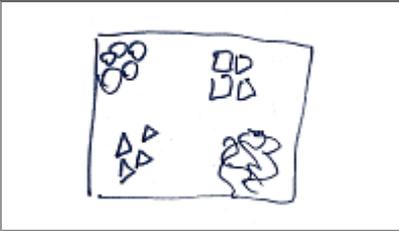
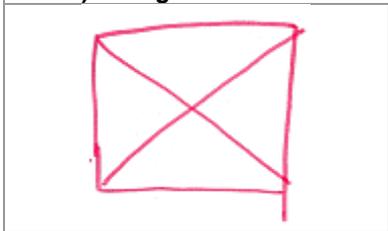
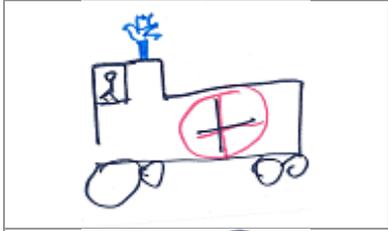
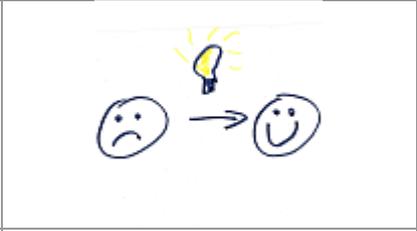
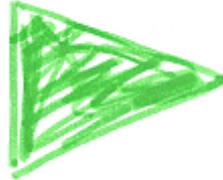
m) Aufgabe nicht geschafft		
		

Tabelle 30: Symbole der Kinder aus Gruppe 3

A6.4 Eigene Symbole

a) Aufgabenauswahl		
		
		
b) Programm beenden		
		
c) Hilfe		
		
		

d) Einen Schritt vorwärts gehen		
		
e) Einen Schritt rückwärts gehen		
f) Nach links drehen		
		
	Symbol für f und g	
g) Nach rechts drehen		
		
h) Start		
		
i) Stopp		
		
j) Löschen		
		

k) Pause		
		
l) Aufgabe geschafft		
		
m) Aufgabe nicht geschafft		

Tabelle 31: Eigene Symbole

A7 Card Sorting

Kategorie	Bedeutung	Symbolnr.	richtig zugeord.	falsch zugeord.	nicht zugeord.	Beliebtheit
a	Aufgabenauswahl	49	3	0	2	1
a	Aufgabenauswahl	48	2	1	2	2
a	Aufgabenauswahl	47	1	0	4	0
a	Aufgabenauswahl	51	1	0	4	0
a	Aufgabenauswahl	50	0	3	2	2
a	Aufgabenauswahl	28	0	2	3	0
b	Beenden	30	1	3	1	4
b	Beenden	3	1	1	3	1
c	Hilfe	44	5	0	0	3
c	Hilfe	1	1	1	3	0
c	Hilfe	43	1	2	2	0
c	Hilfe	41	0	2	3	2
c	Hilfe	5	0	1	4	0
c	Hilfe	40	0	1	4	0
c	Hilfe	20	0	2	3	0
c	Hilfe	39	0	2	3	0
d	Vorwärts	36	5	0	0	3
d	Vorwärts	13	4	0	1	5
d	Vorwärts	22	4	0	1	1
d	Vorwärts	6	1	0	4	2
f	Links	31	3	2	0	2
f	Links	4	1	1	3	3
g	Rechts	35	3	2	0	4
h	Start	29	5	0	0	1
h	Start	16	4	1	0	3
h	Start	42	1	2	2	1
h	Start	18	1	3	1	0
h	Start	21	0	1	4	0
h	Start	46	0	4	1	0
i	Stopp	11	4	0	1	3
i	Stopp	34	4	1	0	2
i	Stopp	17	1	1	3	0
i	Stopp	12	0	5	0	0
j	Löschen	53	5	0	0	4
j	Löschen	19	4	1	0	1
j	Löschen	52	4	0	1	0
j	Löschen	54	4	0	1	0
j	Löschen	2	1	1	3	0
j	Löschen	26	0	1	4	0
j	Löschen	23	0	3	2	0
k	Pause	45	5	0	0	0

Kategorie	Bedeutung	Symbolnr.	richtig zugeord.	falsch zugeord.	nicht zugeord.	Beliebtheit
k	Pause	14	4	1	0	0
k	Pause	24	1	2	2	5
k	Pause	15	0	2	3	0
k	Pause	7	0	4	1	0
l	geschafft!	27	4	0	1	3
l	geschafft!	37	4	1	0	1
l	geschafft!	38	4	1	0	0
l	geschafft!	9	3	2	0	1
l	geschafft!	8	1	4	0	0
m	nicht geschafft!	32	4	1	0	3
m	nicht geschafft!	25	4	1	0	1
m	nicht geschafft!	33	4	1	0	1
m	nicht geschafft!	10	2	3	0	0

Tabelle 32: Konsolidierte Ergebnisse Card Sorting und Bewertung nach 5 Kindern

A8 Hinweise zur Kodierung Evaluation Prototyp 1

Die Kodierung erfolgt analog zu den bisherigen Kodierungen: „X“ wird für trifft zu kodiert, „-“ für trifft nicht zu, „n.g.“ für nicht gefragt, „n.v.“ Szene kam nicht vor bzw. hat nicht stattgefunden, Kind hat nicht geantwortet oder die Testleitung hat die Aufgabenstellung für das Kind gelöst. Kategorien, aus denen immer nur eine Kategorie (Single Choice) mit „X“ kodiert werden kann, sind als Gruppe auf der rechten Seite mit einem Doppelbalken gekennzeichnet.

Bei der Usability-Evaluation legt die Verfasserin der gegenständlichen Arbeit besonderes Augenmerk auf eine selbsterklärende Benutzeroberfläche des Prototyps 1. Daher ist, sofern nicht anders angegeben, immer das erste Vorkommen von Ereignissen oder Aktionen zu kodieren, damit ersichtlich wird, ob die Kinder ohne Anleitung und Einweisung mit der Software arbeiten könnten oder nicht.

Die Kategorie PT101 ist beim ersten Auftreten des Ereignisses bzw. der Aktion zu kodieren. Es ist zu vermerken, ob das Kind selbständig (nach einer Aufforderung durch die Testleitung) eine Aufgabe auswählen kann oder ob es dazu Hilfestellungen benötigt (PT10101, PT10102, PT10103). Wenn das Kind nach dem Aufruf der Aufgabe nur die Mitte des Bildschirms im Fokus hat und Details beschreibt, dann ist dies in Kategorie PT10104 zu kodieren. Ebenso in dieser Hauptkategorie ist zu erfassen, ob das Kind den grünen Bereich, der die Aufgabenstellung beinhaltet, von sich aus erkennt (PT10105) und die Aufgabenstellung in eigenen Worten wiedergeben kann (PT10106) sowie den Button zur Aufgabenauswahl findet (PT10107).

Die zweite Kategorie, PT102, beinhaltet das Erkennen und Auffinden der zentralen Bedienelemente der Benutzeroberfläche. Findet das Kind selbständig, auf Anhieb und ohne Hilfestellung

- den Navigationsbereich (PT10201),
- den Button zum Losfahren, Start (PT10202),
- den Button zum Löschen eines Befehls (PT10203),
- den Button zum Beenden des Programms (PT10206) und
- den Button für Pause (PT10207).

Weitere interessante Aspekte für die Beobachtung sind, ob das Kind von sich aus die Zeitanzeige thematisiert (PT10204) oder auf Nachfrage die Zeitanzeige als Zeitanzeige bzw. Fortschrittsanzeige wahrnimmt (PT10205). Erkennt das Kind die Bedeutung des Doppelpfeils (bei Aufgabe nicht geschafft, PT10208) und kann bei erfolgreich gelöster Aufgabe und dem Erscheinen von „Super!“ zur Aufgabenauswahl wechseln, in dem es auf „Super!“ klickt (PT10210) oder auf einen anderen Bereich des Bildschirms tippt (PT10211).

Hilfreich ist die Befehlsliste, in der alle eingegebenen Befehle angezeigt werden. Erkennt das Kind diese ohne Hinweise (PT10209)? Zuletzt zu beobachten ist, ob das Kind Stopp drückt und dadurch irritiert wird oder Stopp mit Start verwechselt (PT10212).

Die Kategorie 3, PT103, beschäftigt sich mit dem Thema der Hilfe. Das Programm stellt über den Button mit dem Rettungswagen einen Hilfebildschirm zur Verfügung. Mit PT10301 ist zu erfassen, ob das Kind auf Anhieb und ohne Hilfestellung den Button für die Hilfe findet. Wenn das Kind beschreiben oder erklären kann, was der Hilfebildschirm darstellt, ist dies in Kategorie PT10302 zu kodieren. Sollte die Hilfe das Kind irritieren, ist das in Kategorie PT10303 zu dokumentieren und wenn das Kind ohne Hilfestellung wieder zurück zur Aufgabe oder Aufgabenauswahl wechseln kann, ist das in Kategorie PT10304 zu vermerken.

Die nächste Kategorie, PT104, berücksichtigt die sinngemäße Nutzung der durch den Prototyp repräsentierten Software. In den jeweiligen Unterkategorien ist zu erfassen, ob das Kind den Start-Button bewusst sinngemäß nutzen kann (PT10401), ob das Kind die

Richtungs-Buttons bewusst sinngemäß nutzen kann (PT10402) und ob es eine erste einfache Aufgabenstellung, in der Regel die Spielfigur einen Schritt oder zwei Schritte vorwärts bewegen, ohne Hilfestellung fahren kann (PT10403).

In weiterer Folge stellte die Testleitung den Kindern eine komplexere Aufgabe. Hierbei ist es von Interesse, ob das Kind die Aufgabenstellung ohne Hilfestellung (PT10404), mit Hilfestellung (PT10405) oder mit erheblicher Hilfestellung (PT10406) lösen kann.

Zum Abschluss sind noch Kategorien zur Erfassung der Begeisterung des Kindes (PT105) zu kodieren. Will ein Kind von sich aus weitere Aufgaben lösen, so ist das in Kategorie PT10501 abzubilden. Findet das Kind den TouchScreen bzw. die Eingabe via TouchScreen (PT10502) oder die Buttongröße (PT10503) gut, sind die entsprechenden Kategorien zu kodieren. Wenn der Schmetterling als Spielfigur Anklang findet (PT10504), der hier verwendete Schmetterling gut gefällt (PT10505) oder die Farbgestaltung des Arbeitsbereichs für das Kind ansprechend gewählt wurde (PT10506), ist das in den jeweiligen Kategorien zu repräsentieren.

A9 Hinweise zur Kodierung Evaluation Prototyp 2

Die Kodierung der Beobachtungskategorien erfolgt analog zur Kodierung der Beobachtungskategorien der Usability-Evaluierung des Prototyps 1 (siehe Kapitel A8).

Ergänzend ist anzumerken, dass die Lösung der Aufgaben 1, 2 und 3 durch die Kinder entsprechend dem Grad der Hilfestellung zu erfassen ist. (PT20403 bis PT20411). Die Kategorie PT20506 ist zu kodieren, wenn das Kind nach eigenen Angaben das Programm schön oder Gefallen daran findet. Möchte das Kind mit dem Programm spielen (wenn es dieses gäbe bzw. wenn das Kind die Möglichkeit hätte, dass das Programm bei ihm am Computer installiert wäre), dann ist das in Kategorie PT20507 entsprechend zu vermerken sowie Kategorie PT20508 zu kodieren, wenn das Kind der Meinung ist, dass es mit dem Programm auch allein ohne Hilfestellung spielen kann.

Lebenslauf

Mag.rer.nat. Kerstin Stöckelmayr schrieb ihre Dissertation am Institute of Visual Computing and Human-Centered Technology - Forschungsbereich Human Computer Interaction an der TU Wien.

Ausbildung

Universität

2000 – 2006 Studium Informatik und Mathematik Lehramt an der Universität Wien und Technischen Universität Wien, Abschluss: Mag.rer.nat.

Aus- und Weiterbildungen in den Bereichen

Projektmanagement,
Requirements Engineering,
Lern-Coaching,
Qualitätsmanagement,
Gender Mainstreaming und Diversity

Beruflicher Werdegang

2005 – 2006 E-Learning Mitarbeiterin bei e-Lisa academy
2007 – 2010 Stadtschulrat Wien
2010 – 2011 Wissenschaftliche Mitarbeiterin Robotics in Education,
FH Technikum Wien
2010 – heute Selbständige Unternehmerin
2011 – heute Lektorin, FH Wr. Neustadt
2013 – heute Lektorin, FH Technikum Wien

Publikationen

Papers

Stöckelmayr, Kerstin; Tesar, Michael; Hofmann, Alexander (2011) Kindergarten Children Programming Robots: A First Attempt In: Proceedings of 2nd International Conference on Robotics in Education (RiE 2011). Vienna, Austria, September, 2011, S. 185 - 192. INNOC - Austrian Society for Innovative Computer Sciences, Web: <http://www.rie2011.org/conference/proceedings>

Tesar, Michael; *Stöckelmayr, Kerstin*; Sieber, Stefanie; Pucher, Robert (2011) Agilität als Chance zum Qualitätsmanagement in modernen Lehr-Lern-Szenarien In: Thomas Köhler, Jörg Neumann (Hrsg.) Wissensgemeinschaften, Digitale Medien – Öffnung und Offenheit in Forschung und Lehre, 2011, 16. Jahrestagung der GMW, Buchreihe Medien in der Wissenschaft, Band 60, 408 Seiten, Waxmann Verlag, S. 157 - 167, ISBN 978-3-8309-2545-3

Salzbrunn, Benedikt; Tesar, Michael; *Stöckelmayr, Kerstin*; Pucher, Robert (2011) Early Software Prototyping for Usability Tests Using Low Cost Tools for Teaching Purposes In: Proceedings of the International Technology, Education and Development Conference (INTED) 2011, CD, Valencia, Spain, March 7th – 9th 2011, ISBN: 978-84-614-7423-3

Pucher, Robert; Salzbrunn, Benedikt; Tesar, Michael, *Stöckelmayr, Kerstin* (2011) Mobile Learning Apps and Educational Web-Platforms for Organizational Aspects of Teaching In: Proceedings of EDULearn11, CD, July 4th-6th, Barcelona, Spain, ISBN: 978-84-615-0441-1

Stöckelmayr, Kerstin; Tesar, Michael; Hofmann, Alexander; Pucher, Robert; Schmöllebeck, Fritz (2011) New Role Models for IT-Students to Improve their Motivation to Learn In: T. Bastiaens & M. Ebner (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011, S. 2318 - 2323. Chesapeake, VA: ACE

Pohl, Margit; Rester, Markus; Judmaier, Peter; Reichl, Franz; Jerlich, Jutta; Obermüller, Eva; *Stöckelmayr, Kerstin* (2006) Designing Communication and Cooperation Processes for Adult Learners In: M. Jones, A. Krieger, F. Reichl, A. Steiner (Eds.), Proceedings of the 10th IACEE World Conference on Continuing Engineering Education (WCCEE) 2006, Vienna, Austria, S. 1 - 5, ISBN: 3-85288-200-1

Buchkapitel

- Tesar, Michael; *Stöckelmayr, Kerstin*; Pucher, Robert; Ebner, Martin; Metscher, Johannes; Vohle, Frank (2013) Multimediale und interaktive Materialien – Gestaltung von Materialien zum Lernen und Lehren In: M. Ebner & S. Schön: L3T – Lehrbuch für Lernen und Lehren mit Technologien, 2. Auflage, Web: <http://l3t.eu>, ISBN: 978-3-8442-6594-1
- Zorn, Isabel; Trappe, Christoph; *Stöckelmayr, Kerstin*; Kohn, Tanja; Derndorfer, Christoph (2013) Interessen und Kompetenzen fördern – Programmieren und kreatives Konstruieren In: M. Ebner & S. Schön: L3T – Lehrbuch für Lernen und Lehren mit Technologien, 2. Auflage, Web: <http://l3t.eu>, ISBN: 978-3-8442-6594-1
- Tesar, Michael; Pucher, Robert; *Stöckelmayr, Kerstin*; Metscher, Johannes; Vohle, Frank; Ebner, Martin (2011) Interaktive, multimediale Materialien - Gestaltung von Materialien zum Lernen und Lehren In: M. Ebner & S. Schön: L3T - Lehrbuch für Lernen und Lehren mit Technologien, ISBN: 978-3-8442-0437-7, Web: <http://l3t.eu>
- Pohl, Margit; Rester, Markus; Judmaier, Peter; Reichl, Franz; Jerlich, Jutta; Obermüller, Eva; *Stöckelmayr, Kerstin* (2005) Ecodesign - ein Online-Universitätslehrgang für nachhaltige Produktentwicklung In: Forschung zu Blended Learning: österreichische F & E Projekte und EU-Beteiligungen, issued by: Forum Neue Medien, Graz, Österreich, S. 181 - 192, ISBN: 3-902520-01-9

Journalbeiträge

- Pohl, Margit; Rester, Markus; *Stöckelmayr, Kerstin*; Jerlich, Jutta; Judmaier, Peter; Reichl, Franz; Obermüller, Eva (2008) Computer Supported Collaborative Learning and Vocational Training - Adapting the Technology to the Learners' Needs In: Universal Access in the Information Society, Vol. 7 (2008), No. 4, S. 259 – 272
- Pohl, Margit; Rester, Markus; Judmaier, Peter; *Stöckelmayr, Kerstin* (2005) Ecodesign – Design and Evaluation of an E-Learning System for Vocational Training In: Elektrotechnik und Informationstechnik (e&i), Vol. 122 (2005) No. 12, S. 473 – 476