TU WIEN Informatics

Analyse, Design und prototypische Implementierung eines Serious Game zur Balance-Rehabilitation mit Machine Learning-basierter, personalisierter Levelgenerierung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Johannes Riedmann, BSc
Matrikelnummer 0926649

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dipl.-Ing. Dr.techn. Thomas Grechenig
Mitwirkung: Dipl.-Ing. Dr.techn. René Baranyi

Wien, 25. August 2022

Unterschrift Verfasser

Unterschrift Betreuung

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Analysis, design and prototypical implementation of a serious game for balance rehabilitation providing personalized level generation based on machine learning techniques

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Johannes Riedmann, BSc
Registration Number 0926649

to the Faculty of Informatics

at the TU Wien

Advisor: Dipl.-Ing. Dr.techn. Thomas Grechenig
Assistance: Dipl.-Ing. Dr.techn. René Baranyi

Vienna, 25th August, 2022

Signature Author

Signature Advisor

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Analyse, Design und prototypische Implementierung eines Serious Game zur Balance-Rehabilitation mit Machine Learning-basierter, personalisierter Levelgenerierung

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Johannes Riedmann, BSc
Matrikelnummer 0926649

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Dipl.-Ing. Dr.techn. Thomas Grechenig

Wien, 25. August 2022

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Johannes Riedmann, BSc

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.

Wien, 25. August 2022

Johannes Riedmann

Acknowledgements

First and foremost I would like to thank my wonderful parents, Thomas and Brigitte, as well as my friends for supporting me throughout my academic career. I would also like to thank René Baranyi, member of the research group for industrial software and my thesis advisor, for his valuable insight and continuous guidance. Furthermore, I would like to thank Raphaela Borg and Hanna Schlosser, physiotherapists at aks Vorarlberg and SMO Dornbirn respectively, for sharing their expert knowledge with me during interviews and the evaluation phase of the prototype. At this point, I also want to express my gratitude to my employer, Zühlke Engineering (Austria) GmbH, and especially Barbara Hotwagner for granting me educational leave to finalize my masters studies. Finally, I would like to thank my fellow student and good friend Felix Schuller for his consistent drive and dedication throughout our shared studies.

Kurzfassung

Eingeschränkte Balancefähigkeit zeigt sich besonders häufig in der älteren Bevölkerung, kann aber auch als Resultat eines Unfalls oder einer Krankheit auftreten. Wenn Balanceprobleme verherend genug werden und Einfluss auf das alltägliche Leben des Betroffenen haben müssen Rehabilitationsmaßnahmen gesetzt werden. Basierend auf dem Grad an Beeinträchtigung kann Rehabilitation mehrere Jahre in Anspruch nehmen, was Einfluss auf die Motiviation des Patienten haben kann. Serious Games (SG, Engl. für "Spiele mit ernsthaftem Ziel") haben sich als wertvolle Ergänzung zur konventionellen Balance-Rehabilitation herausgestellt indem sie Motivation und Engagement der Patienten, auf ihrem Weg zu einem unabhängigeren Lebensstil, verstärken.

Diese Arbeit beschreibt die Analyse und Entwicklung eines neuen SG für Balance-Rehabilitation. Zur Datensammlung und -auswertung der Gewichtsverlagerungsübungen während des Spiels wird das WBB verwendet, welches eine günstige Alternative für eine Kraftmessplatte darstellt. Der Prototyp wird auf Basis von Anforderungen entwickelt, die während einer umfangreichen Literaturrecherche und Interviews mit zwei Physiotherapeuten gesammelt wurden, um den aktuellen Stand der Technik bezüglich Designkriterien von SG für Balance-Rehabilitation, sowie die Anwendung des Prototypen im klinischen Umfeld zu eruieren.

Im resultierenden Prototyp mit dem Namen *Walk in the park* (Engl. für "Spaziergang im Park"), absolvieren Spieler Levels, indem sie mittels Gewichtsverlagerungsübungen auf dem WBB einem Pfad entlang navigieren. Ein essentieller Aspekt beim Design für SGs ist die Anpassung der Spielschwierigkeit basierend auf den Einschränkungen der Patienten. Im aktuellen Stand der Technik wird Balancefähigkeit durch erreichte Spielerfolge angenähert, dementsprechend leitet sich die Spielschwierigkeit von diesen abstrakten, spielbezogenen Erfolgen ab. *Walk in the park* stellt einen neuen Ansatz zur Anpassung der Spielschwierigkeit vor, bei dem die Daten, aufgezeichet und gesendet vom WBB während des Spiels, genutzt werden um personalisierte Levels zu generieren, basierend auf einer datenzentrierten Balanceauswertung. Ein weiterer Anforderungskatalog, entsprungen aus dem Interviewprozess, umfasst notwendige Aspekte, um durch Levelgenerierung auf die Einschränkungen von Patienten einzugehen. Diese Anforderungen finden sich in der personalisierten Levelgenerierung des Prototypen wieder. Desweiteren wird in dieser Arbeit ein neuartiger Ansatz zur Quantifizierung von Balancefähigkeit, basierend auf WBB-Daten, beschrieben. Hierbei werden zunächst Daten von Menschen ohne Balanceeinschränkungen gesammelt, während diese Gewichtsverlagerungsübungen ausführen, um

daraus ein Modell für Balancefähigkeit in gesunden Menschen abzuleiten. Anschließend wird mittels Distanzmetriken die Abweichung neuer Messungen zum zugrundeliegenden Modell gemessen, woraus eine Balanceauswertung gebildet werden kann.
Walk in the park wurde in vier Rehabilitationssitzungen mit beeinträchtigten Patienten eingebunden unter strenger Beobachtung der teilnehmenden Physiotherapeuten. Die Evaluierungsergebnisse zeigen, dass *Walk in the park* ein motivierendes und spannendes Spielkonzept bietet. Desweiteren reflektieren die erreichen Balanceauswertungen die Einschränkungen der Patienten, was darauf hindeutet, dass der entwickelte Prototyp es ermöglicht generelle Balanceschwäche, sowie Schwächen in direktionalen Gewichtsverlagerungen zu erkennen.

Abstract

Decreased ability in balance control often arises in the aging population, but can also be the result of an accident or disease. When balance impairments are severe enough to negatively affect people's everyday life, balance rehabilitation must be considered. Based on the severity of impairments, rehabilitation can take several years, which might affect patient motivation. Serious Games (SG) have proven to be a valuable addition to balance rehabilitation, boosting patient motivation and engagement during their journey to a more independent lifestyle.

This thesis describes the analysis and development of a new SG for balance rehabilitation. To gather and assess data on the performed weight-shifting exercises during gameplay, the WBB, a low-cost force plate, is utilized. The resulting prototype is developed based on requirements gathered from an extensive literature research and interviews with two physiotherapists, investigating design criteria for state-of-the-art SG in balance rehabilitation and requirements for applying it to clinical practice.

In the resulting prototype, called *Walk in the park*, players complete levels by navigating along a path by performing weight-shifting exercises on the WBB. A vital part in SG design is to adjust game difficulty based on patient impairment. In current state-of-the-art work, balance ability is estimated by the achieved in-game objectives, thus difficulty is adjusted based on abstract achievements. *Walk in the park* introduces a novel approach for difficulty adjustment by leveraging the data stream emitted by the WBB during gameplay to generate personalized levels based on data-centric balance scores. Another set of requirements, gathered during the interview process, summarizes the necessary aspects for level generation to accommodate for patient impairments. These requirements are incorporated into personalized level generation. Furthermore, this thesis describes a novel approach for quantifying balance ability based on WBB data. In essence, data from individuals without balance deficits, performing directional weight shifts, is gathered to build a model that reflects healthy balance ability. A distance measure captures the deviation from a new measurement and the underlying model and assigns a balance score for each directional weight shift.

Walk in the park was integrated into four balance rehabilitation sessions with balance-impaired patients under close supervision of the participating physiotherapists. Evaluation results suggest that *Walk in the park* provides motivating and engaging game design and mechanics. Furthermore, achieved balance scores reflect patient balance impairments, which indicates that the developed prototype is capable of detecting overall balance

impairment and weaknesses in directional weight shifts.

Contents

CHAPTER 1

Introduction

Balance impairments can arise from a multitude of reasons, including accidents, diseases and old age. In moderately severe cases, where a patient's quality of life suffers from the lack of balance ability, rehabilitation sessions are decreed to increase patient mobility and reduce the risk of falls. Any kind of rehabilitation after an accident, medical intervention or disease can be a cumbersome endeavor and take up to several years. The process of recovery often fails due to a lack of motivation to repeatedly perform a strict routine of tasks and exercises.

Serious Games (SGs) are games, designed to primarily follow an educational or otherwise productive purpose as main stimulus and entertainment as an additional dimension. In the healthcare sector, SGs are used to motivate and engage patients e.g. to document their lifestyle for medical purposes or perform rehabilitation exercises. SGs that involve performing physical or mental exercise are called exergames. Numerous exergames for balance therapy have been developed, but are rather static and do not properly reflect and adjust to the patient's abilities and needs.

The use of software tools utilizing machine learning techniques is steadily increasing in the healthcare sector to support clinical workflows, assessments and decision making. The core concept of machine learning is to derive models that provide valuable information from raw data. This is used to a great extent in medical image processing, medication planning and cohort studies. To gather the necessary data for modeling movement behavior in balance rehabilitation, the sensory data emitted by force plates can be utilized. Furthermore, the extracted data can be applied to adapt game elements to more precisely encompass a patient's physical constraints.

1.1 Problem description and motivation

Balance rehabilitation is a crucial part of recovery following neurological injury [1]. Its main goal is to retrain muscle strength, balance ability and coordination to enable the patient to go about their day-to-day activities without the need for external assistance. Stroke survivors, for example, most prominently suffer from balance problems, which affects patient's everyday life by reducing mobility and increasing their risk of falling [2]. Besides diagnosed neurological injury, old age is a common perpetrator causing cognitive constraints that can result in balance impairment. A third of persons at the age of 65 or older fall at least once a year, where 10 percent of falls result in serious injuries. Besides the physiological effects of a fall, the incident also takes its toll on the person's psychological well-being, causing the victim to move less, fearing another potential fall. This behavior subsequently leads to a decreased level of independence, reduced muscle strength and an increased risk of future falls [3].
If balance impairments caused by old age, severe enough to potentially cause a fall incident, were detected early enough, patients could be subjected to physical therapy as preventative measure. Persons suffering from neurological injuries, on the other hand, have to undergo rehabilitative measures to potentially regain their former balance ability. Regardless of physical therapy resulting from preventative or rehabilitating origins, the process of recovery has to be early, intensive and repetitive, which can result in problems with patient motivation [4].
Integrating video games into the physical therapy process to engage and motivate patients has been proven to be effective, yet there are still drawbacks associated with this in practice. Off-the-shelf video games can be used to augment traditional balance rehabilitation techniques, but they are not designed for therapeutic purposes, thus are often too difficult for patients with moderate or severe balance impairments [5]. Furthermore, a lot of custom-made SGs for rehabilitation purposes require sophisticated hardware (e.g. Virtual Reality games) and considerable setup effort, thus are not feasible for at-home use. Designing and developing new SG from scratch, with the target group of physically impaired patients and minimal setup cost and effort in mind, can be a way to circumvent these limitation.
The Nintendo Wii Balance Board (WBB) is a low-cost force plate that is widely used in the field of SGs in balance rehabilitation and has proven to be a valid replacement for laboratory-grade force platforms [6]. Performing physical exercises on the WBB allows gathering of sensory data and gaining insights about the patient's balance ability [3]. Current state-of-the-art work for SGs in balance rehabilitation adjusts game difficulty based on scores derived from in-game achievements, ignoring the gathered data during gameplay [7][8]. This data is rich in information and can provide insight into patient balance impairments [3][9]. Thus, data gathered during exercise execution on the WBB can be utilized to adapt game difficulty based on balance ability and highly personalize the gaming experience, thus boost patient motivation and engagement for balance rehabilitation.

1.2 Aim of this work

The aim of this thesis is to develop a prototype of a new SG, providing personalized game levels by evaluating the patient's movement behavior, utilizing the WBB as sensory device. A prototype for the SG is developed, based on current state-of-the-art best practices, and evaluated in the process. For the purpose of balance evaluation, machine learning techniques are utilized, comparing movement patterns of balance impaired patients, sampled during gameplay, with baseline data collected from healthy individuals. To achieve this, balance ability of healthy individuals is modeled. Evaluation of patient impairments can be computed by quantifying the divergence from the healthy model. The level generation mechanism incorporates findings from the machine learning-based evaluation stage with data from previous sessions to generate personalized levels. The resulting tailored level design is expected to adequately adapt to the patient's rehabilitation progress and ultimately improve overall patient motivation and engagement.

The thesis aims to answer the following research questions:

1. **What are the requirements for a SG in the context of balance rehabilitation?** The resulting prototype should be applicable in clinical practice, thus integrating features to support established balance rehabilitation workflows is pivotal. Furthermore, gamification techniques for SGs have to be evaluated and integrated. This research question is answered by conducting interviews with physiotherapists, combined with extensive literature research.

2. **What are the requirements for personalized level generation in balance rehabilitation?** This research question aims to answer how a level generation mechanism can be developed to increase patient engagement and motivation to subsequently minimize frustration and exhaustion during rehabilitation. To answer this research question, physiotherapists are interviewed regarding balance rehabilitation techniques, therapy workflows and ways to incorporate the gathered information into personalized level generation.

3. **Can machine learning techniques be utilized to derive personalized game levels from patient movement patterns extracted from a WBB?** From a technical perspective, this question answers how game level generation based on WBB sensory data is achieved. More precisely, how data is sampled from a WBB sensory data stream, how useful features can be derived to detect weaknesses in balance ability and how these features are used in combination with machine learning techniques to parameterize level generation. A list of acceptance criteria to quantify conformity and correctness of level generation performance is compiled and evaluated by a physiotherapist to assess this research question.

1.3 Methodology

The methodological approach in this thesis encompasses three phases, including a research phase, an implementation phase and, finally, an evaluation phase. Figure 1.1 associates each phase with the participating stakeholders and visualizes relative duration and chronological order of steps within the phases. Stakeholder descriptions can be found in section 4.1.1.

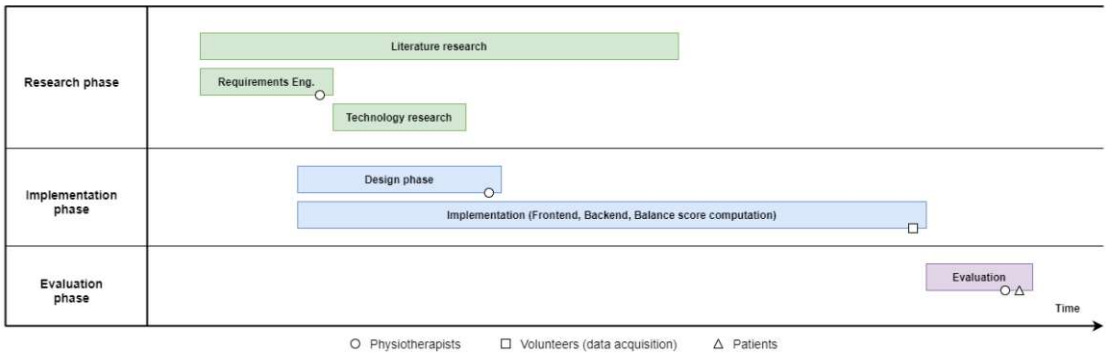


Figure 1.1: Flow chart associating phases with participating stakeholders

In the following, each of these phases and the expected outcomes are outlined.

Research phase The first step during the research phase is requirements engineering. This phase is pivotal in answering the first two research questions regarding requirements of SGs and personalized level generation in balance rehabilitation. During this phase, two physiotherapists were questioned regarding clinical routine and expectations management. In these interviews, mechanisms to support clinical workflows that have to be incorporated into the level generation mechanism and gameplay were defined. A mutual understanding of the expected results was established and requirements for gameplay, as well as level generation, were defined. The interviews resulted in two individual requirements catalogs. One of the catalogs is focused on requirements for usage of the prototype in clinical practice, whereas the second catalog covers requirements regarding personalized level generation.
In an overarching literature research phase, insight gained from the requirements engineering phase and current state-of-the-art work regarding SGs in balance rehabilitation was analyzed and evaluated in greater detail, utilizing academic search engines, e.g. Google Scholar. As a result, another catalog containing vital design criteria for SGs in balance rehabilitation was compiled. Furthermore, research was conducted regarding feasibility of evaluating patient balance ability in a way that enables a level generation mechanism to create personalized levels. This step is vital in answering the third research question. Furthermore, decisions regarding the technological approach were made. Thus, a technical research phase was conducted, evaluating system architecture, required components and

interactions, as well as utilized frameworks, technologies and programming languages. System architecture and component interaction was modeled via UML.

Implementation phase In the implementation phase, the prototype of the SG was developed, based on information gathered in the previous steps, using rapid prototyping. To ensure that requirements are met during development, mock-ups were designed and discussed with both physiotherapists in an iterative manner during in the scope of a designated design phase. The implementation phase provides further insight regarding the third research question. This phase resulted in a novel SG, called *Walk in the park*, generating personalized levels based on the player's movement behavior that respects the gathered requirements.

Evaluation phase Finally, in an evaluation phase, the proposed system was integrated into physical rehabilitation sessions for patients with balance impairments under close supervision of physiotherapists. The prototype was evaluated in regards to fulfillment of requirements defined via the first two research questions. Furthermore, this phase provides insight into the proposed system's capability of generating personalized levels for the patients and how well the prototype is received.

1.4 Structure of the thesis

Chapters 2 and 3 cover the theoretical background, where chapter 2 focuses on discussing the core domains relevant to the thesis in greater detail and chapter 3 dives into the current state of the art regarding SG in balance rehabilitation. The results of this thesis, subdivided in sections dedicated to the respective phases of development, are presented in chapter 4. Results of the evaluation phase are described in chapter 5. A discussion of the results and possible future work can be found in chapters 6 and 7 respectively.

CHAPTER 2

Theoretical background

To create a general understanding of the underlying context of this thesis, this chapter covers topics relevant to generating personalized levels for SGs in balance rehabilitation. For this purpose, SGs are explained and addressed in the context of healthcare and, more concretely, balance rehabilitation. Subsequently, the role of the WBB in balance rehabilitation, as well as its technical specifications are discussed. Finally, the utilization of machine learning techniques in balance rehabilitation, its use in combination with the WBB and the potential of generating personalized levels in SGs is discussed.

2.1 Gamification elements of Serious Games

The field of SGs is growing rapidly and finds its way into different application areas, such as training, simulation, education or healthcare [10][11]. SGs can be distinguished from conventional video games by their design objectives, meaning SGs do not have entertainment, enjoyment or fun as their primary purpose [12]. SGs always have a serious underlying objective that they aim to motivate and engage the player on working towards. This objective can be, for example, gaining knowledge on a topic for educational purposes, performing exercises for rehabilitation, learning new skills for professionals of trainees and much more. The primary objective of SGs is to combine a serious purpose with elements and techniques borrowed from video games [10] to ultimately turn an otherwise tedious and potentially long-winded task into an engaging endeavor. Applying game elements to otherwise serious contexts, also known as gamification, has proven to be very effective in terms of user motivation and engagement [13]. These gamification mechanics in SGs aim to drive behavioral change in individuals to achieve an underlying goal.

In their systematic review, *Hervas et al.* [14] compiled gamification elements for behavioral change in the context of SGs and created a taxonomy based on their findings. Furthermore,

they linked each gamification element to a psychological concept explaining their relation to the expected behavioral change, see table 2.1.

Psychological concept	Definition
Self-efficacy	Strength of one's belief in personal aptitudes to reach goals
Cognitive restructuring	Learning process to identify and confront ineffective and disruptive behaviors
Social influence	Social power to change behavior of others in a particular direction
Vicarious learning	Learning that is derived from indirect sources, rather than direct instruction
Shaping	Gradually molding to perform a specific response by reinforcing responses that are similar to the desired response
Nudge theory	Reinforcement and indirect suggestions to try to achieve non-forced expected behavior
Behavioral momentum	Relation between resistance to change and the rate of reinforcement obtained

Table 2.1: Main psychological theories and concepts related to gamification for behavioral change [14]

In order for SGs to have positive impact on the player, gameplay has to be designed in such a way, that the player enjoys the game and is incentivized to continue playing. In the following, gamification mechanics, identified by *Hervas et al.*, are explained in more detail and linked to the corresponding psychological concepts.

Goals *Hervas et al.* [14] define this gamification mechanic as the "*main reason and way of acting based on users' ambitions and efforts*", which means that a SG has to focus on progressing towards a main objective. Elements found in SGs focused on achieving the main goals are *achievements*, *challenges*, *quests*, *levels* and the concept of *aversion*. *Achievements* mark milestones in progressing through the game and are rewarded as motivational tokens. *Challenges* and *quests* are predefined sets of actions the user has to take in order to complete them. They should be challenging the player's abilities and reward the player when successfully completed. *Levels* allow for better control of progress by subdividing sets of complex tasks. The concept of *aversion* is implemented by punishing the player for taking steps that prevent them from progressing towards their goals.
Gamification elements aimed to reach the main objectives satisfy to the concept of *self-efficacy*. Reaching goals in a progressive manner reinforce the player's belief in their abilities. Rewarding correct behavior, e.g. though *achievements*, and punishing negative

progress through elements of *aversion* supports the concept of *cognitive restructuring*. *Social influence* can be obtained by adding functionality that allows comparing the goals reached with those of others. The concept of *shaping* is applied by proposing progressive and incremental goals that steadily elevate the player towards their skill ceiling. Creating game mechanics that aim to create a habit, e.g. daily quests or challenges, fall into the psychological concept of generating *behavioral momentum*.

Status This gamification element has the purpose of differentiating players based on characteristics related to their progress within the game. *Hervas et al.* [14] propose *rankings*, *leaderboards* and *social sharing* as elements related to status. *Rankings* show the user's progress relative to their peers and assign a global rank to their name. This establishes a clear hierarchy and motivates players to elevate their status within the playerbase. Showing the leading competitors in a *leaderboard* has the potential to motivate players reaching these ranks. Both *rankings* and *leaderboards* are tools of *self-efficacy*. They help defining new goals by reaching for a specific rank or being part of the elite players on the *leaderboard*. The positive *social influence* a player potentially receives after sharing their in-game progress with an external community motivates the player to continue following their goals.

Randomness Elements of randomness can help make a game more enjoyable by introducing characteristics that make the game seem unpredictable. Rankings and leaderboards are rather static measures of game progress. When the player is either stuck at a specific rank or the point discrepancy between ranks is large, motivation to continue playing might suffer because there is no easily obtainable reward. Introducing elements of randomness to the game can help boost motivation. *Hervas et al.* [14] have found *free lunch*, *luck*, *variations* and *surprise* to be the most prominent gamification elements in SGs. *Luck*, *free lunch* and *surprise* are elements that randomly confront the player with unexpected rewards, e.g. bonus points or hidden in-game content. Implementing *variations* make a game less predictable and the player has to adapt more to unknown situations, e.g. parts of level-generation being randomized. Breaking the routine by introducing randomness is supported by the psychological concepts of *nudge theory* and *behavioral momentum*. Randomizing elements of gameplay *nudges* the player to further explore the game world and motivates by handing out occasional rewards at random.

Appointment In the taxonomy proposed by *Hervas et al.* [14], appointments are gamification elements that constraint the player to a predetermined time frame. *Cooldowns* can be used to give the player a maximum amount of time in which a challenge, quest or level has to be fulfilled. This incentivizes the player to grow more accustomed to the game mechanics, thus ultimately become better at the game. *Scheduled events* allow the player to participate in predetermined actions within the game and also create a sense of urgency. It can be used to make interacting with the game a habit. *Nudge theory* and *behavioral momentum* are the foundational psychological concepts behind

this gamification element. The player is lured into making the game a habit, which constitutes a change of behavioral momentum, by continuous nudging.

Scoring The player has to be informed about the progress they are making in the game, which is done with scoring mechanics. *Points*, *combos* and *bonuses* are elements found in SGs for scoring. *Points* are rewarded for completing a single action, e.g. a level, and are a means to measure performance. *Combos* are rewarded for completing multiple subsequent actions and can potentially yield *bonuses*. *Bonuses* can also be awarded for any other exceptional performance by the player[14].
Scoring is closely related to *self-efficacy*, because gaining points and climbing the rankings strengthens the player's belief to reach their goals. Sharing their progress (*social influence*), quantitatively measured by their scoring, and gaining feedback can positively influence their motivation to continue playing the game. Scoring also indirectly incentivizes competitive nature (*nudge theory*), which can also influence motivation and engagement.

Immersion This gamification element is described by *Hervas et al.* as "*the deep mental involvement in something in the gamified context*" [14]. A player within a SG can have a *role* that assigns specific functions and responsibilities to them. Interacting with the in-game world within the bounds of a predefined *role* rewards actions that fulfill the assigned purpose (*Shaping*). Projecting the player themselves as avatar with a given role contributes to self-efficacy, "*because a projection of oneself can make us more objective*" [14]. Painting a *narrative* that defines the story and binds together the components of the game is another element that enhances immersion. Designing a SG with the element of *exploration* in mind can also increase immersion. This can be achieved by introducing collectible items or designing a multifaceted in-game environment. Interacting with a virtual world, given a *narrative* and a well-defined *role* can help identify and counteract negative behavior (*cognitive restructuring*). Creating a framework, or an in-game world and *narrative*, to reinforce positive responses instead of directly communicating the expected behavior can be attributed to the psychological concept of *vicarious learning*.

These gamification mechanics have been used in the gaming industry for centuries to keep players interested and make the game more enjoyable. Designing SGs using these mechanics focusing on serious tasks, such as education, training or rehabilitation have also shown to make them more enjoyable and increase motivation and engagement.

2.2 Application areas of Serious Games in healthcare

Just as it is the case with SGs in general, SGs in healthcare are on the rise and the number of SGs designed and implemented for a variety of purposes within the healthcare sector is steadily increasing [15]. One factor for the popularity of SGs in healthcare is the fact that medical training does not suddenly halt after graduation from school or university. Healthcare professionals are constantly challenged to keep up-to-date with innovative new approaches in patient care and subsequently have to reiterate on already

learned techniques and procedures to be able to provide the highest possible standard of care.
Besides indirectly benefiting from the improvement of care by trained professionals, patients can directly profit from the uprise of SGs, interacting with entertaining games that focus on assessment, rehabilitation or therapy of the patient's illness. Due to increasingly older populations in developed countries, age-related medical conditions such as cognitive impairments and balance problems are destined to become more prevalent [16]. As a result, healthcare systems will be unable to apply the same standard of care for the rising number of chronically ill patients [16], thus alternative treatment methods have to be explored that relieve the healthcare sector, yet provide adequate care. A potent way to unburden the healthcare sector could be to shift responsibility for therapy further towards the patient by implementing sophisticated at-home rehabilitation measures. At-home rehabilitation bears the potential to be effective for a wide range of chronic diseases [17][18], yet patients often lack the motivation to perform exercises at the necessary frequency and intensity. Numerous studies suggest that SGs are capable of motivating and engaging patients in their treatment at home [19][20], thus SGs have the potential to contribute in unburdening the healthcare sector.

The vast number of application areas, benefactors and techniques available to design and implement SGs for the healthcare sector raise the necessity to identify and classify SGs in health by a common taxonomy. Numerous taxonomies for SGs are proposed in literature [10][12], but most are too general and attempt to classify SGs as a whole rather than SGs in healthcare. *Sawyer et al.* [21] propose a taxonomy for games in health, defining prevention, therapy, assessment and education as main areas of health activity. To establish a better understanding of the range of application SGs have in healthcare, in the following, the application areas defined by *Sawyer et al.* are explained and associated with examples for SGs within the context of the respective application area.

Preventative SG Prevention-based SGs are designed with focus on a risk group for a certain disease or disability, e.g. older adults, obese people or smokers. They aim to improve the player's behavior, thus invoke behavioral change, towards a healthier lifestyle, improved quality of life and ultimately prevent potential disease.

SG for therapy The term *rehagames*, referring to SGs for rehabilitation, is often used synonymous for therapeutic utilization of SGs. Accidents or diseases can leave patients with severe disabilities that negatively impact their ability to manage everyday life activities independently. The road to recovery usually is taxing, because effective rehabilitation must be early, intensive and repetitive [4]. Thus, facilitating patient motivation and engagement during rehabilitation is crucial and SGs evidently have a positive impact on patient encouragement and consequentially increased rehabilitation success[22][23].

SG for assessment Medical assessment aims to evaluate the presence or degree of impairment a patient exhibits regarding a specific disease or disability.

SG for education SGs have a wide range of application in medical education, from first aid training to surgical simulations. Continuous training is essential in the medical field and SGs can help achieve this goal with lower cost [24].

Examples for each of these application areas are discussed in more detail in section 3.1.

2.3 Estimation of balance ability

In order to decide if physical therapy is needed or to which extent, balance ability must be assessed first. To measure a patient's initial physical constitution or therapy success, a variety of balance tests have been proposed over the years, such as the *Timed "Up and go" test*, the *Functional Reach test* or the *Berg Balance Scale* (BBS), with the latter being the most prominent and widely-used one [26][9][27]. The following sections describe the most commonly used balance estimation protocols in more detail.

2.3.1 Berg Balance Scale

The BBS is an exercise protocol that consists of 14 balance exercises, see table 2.2, and aims to quantify people's balance ability.

Category	Description	Score
Sitting balance	Sitting unsupported	0-4
Standing balance	Standing unsupported	0-4
	Standing with eyes closed	0-4
	Standing with feet together	0-4
	Standing on one foot	0-4
	Turning to look behind	0-4
	Retrieving object from floor	0-4
	Tandem standing	0-4
	Reaching forward with an outstretched arm	0-4
Dynamic balance	Sitting to standing	0-4
	Standing to sitting	0-4
	Transfer	0-4
	Turning 360 degrees	0-4
	Stool stepping	0-4
Total		0-56

Table 2.2: Components of the BBS [28]

For each exercises between 0 and 4 points can be reached, with a score of 0 meaning the patient was not able to complete the task and a score of 4 representing independent exercise completion [27]. With a total of 56 points, a score of 0-20 suggests moderate to mild balance impairment. Reaching between 21 and 40 points is associated with acceptable mobility and 41+ represents good balance ability and a low risk of falling [27]. Based on the achieved BBS score, further decisions regarding physical therapy can be made.

2.3.2 Timed "Up and Go" Test

During this test, subjects have to stand up from a sitting position, walk three meters, turn, walk back and sit down again without any support from the clinician or devices [29]. The whole process is timed, usually using a stopwatch. The time needed to perform the exercise constitutes the quantitative measure of balance ability produced by this test protocol. Further deductions about balance ability can be made by the physiotherapist by observing patient movement during the exercise.

2.3.3 Functional Reach Test

This test assesses the maximum distance that a subject can reach forward from a standing position while being able to maintain a fixed base of support. To perform the Functional Reach Test, subjects have to make a fist and raise their arms to 90 degrees of flexion. Subsequently, the subject is asked to reach as far forward as possible without losing balance. This process is repeated three times. The functional reach is the mean difference of reach between the normal standing position and the maximum forward position while still maintaining postural control [29].

2.3.4 Dynamic Gait Index

The Dynamic Gait Index (DGI) consists of eight exercises: (1) walking, (2) walking while changing speed, (3) walking while turning the head horizontally (4) and vertically, (5) walking with pivot turn, (6) walking over (7) and around obstacles and (8) stair climbing [30]. For each item of the DGI protocol, the subject can score between 0 and 3 points. A score of 0 corresponds to severe impairment and 3 means normal balance ability. The best possible score over the whole exercise catalog is 24 points. The DGI has been found to be a valid and reliable indicator for fall risk, as a score of below 19 has been shown to indicate a increased risk of falling in elderly patients [30].

2.4 Serious Games in balance rehabilitation and the Wii Balance Board

Reduced balance ability vastly impacts a person's psychological and physiological well-being. Affected people suffer from an omnipresent fear of falling, reducing confidence in their ability to pursue their everyday activities. The resulting decreased range of motion

severely impacts the affected person's quality of life. Consequentially, the decrease in physical activity leads to degeneration of muscle strength, thus further amplification of fall risk [3]. When a fall incident happens, especially for elderly people, the road to recovery encompasses extensive physical therapy over the course of months or even years. Conventional physical therapy encompasses a variety of exercises in combination with numerous adaptations to the patient's skill level. Although there is a large pool of exercises and variations to choose from, repetition, consistency and perseverance are the most important factors for rehabilitation success. Guided exercises performed in physiotherapy sessions have to be complemented with at-home training, which is particularly hard for most patients, because of the lack of human supervision and guidance. The rehabilitation procedure can be very taxing and patients often lack motivation to endure extended physiotherapy. SGs have shown to be effective in boosting patient motivation and engagement for otherwise daunting tasks and bear potential to further support patient endurance in their road to recovery from fall incidents [25].
SGs incorporate physical exercises needed for balance rehabilitation into an engaging, gamified context, thus they can be categorized as exergames. The main goals of these games are to create incentives for the player to stick to their exercise routine and perform prescribed exercises in a correct fashion. To evaluate correctness of execution a control mechanism has to be in place. In a gaming context, this control mechanism corresponds to the input device the player uses to interact with the game. A variety of creative ways to navigate through an in-game world of SGs in balance rehabilitation have been proposed in the literature, such as computer-vision-based motion tracking [31], custom-made sensory devices [32] or force platforms [33]. The latter has gained a lot of traction in the field of balance rehabilitation, because compared to computer-vision-based systems, working with force plates as input device reduces complexity regarding system architecture and implementation and bears potential for easier at-home training solutions. Custom-made sensory devices need a lot of expert knowledge to develop and operate, are usually constructed for a specific task and typically more expensive than force plates. Research in SGs for balance rehabilitation using force platforms drastically accelerated with the release of the WBB by Nintendo in 2007, because it was found to be a viable alternative to laboratory-grade force plates [34] at a price around one-hundred dollars. This rise of the WBB in the research community for balance rehabilitation led to the field being referred to as *WiiHabilitation* [5].

2.5 Wii Balance Board in balance rehabilitation

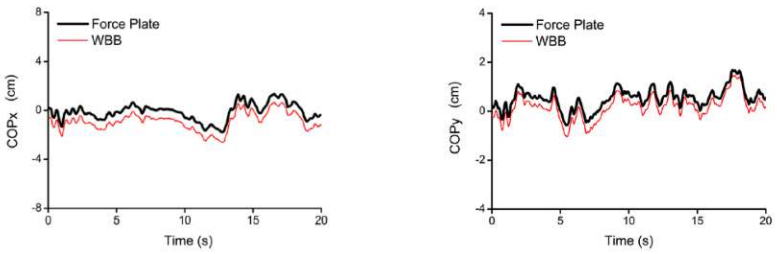
In clinical practice, data from force platforms is used to assess a patient's balance ability. To achieve this, the patient has to perform several physical exercises under varying circumstances, such as quiet standing with eyes closed, while the therapist interprets the resulting data.

2.5.1 Use of force platforms in balance rehabilitation

In their work, *Mansfield et al.* [35] show how data obtained from force plate measurements can be used to reveal patient-specific balance control problems. This information can be used to track the patient's therapy success and to perform necessary adjustments in treatment. Similarly, in their research, *Laufer et al.* [36] investigate postural control of post-stroke patients suffering from hemiparesis, a weakness of one entire side of the body. In their studies, thirty patients underwent functional and posturographic testing on force plates to collect data on postural sway and weight distribution. The aim of their study was to understand the contribution of visual and somatosensory input during exercise, which was calculated based on the center-of-pressure (COP) data retrieved from a force platform.

2.5.2 Validity of the Wii Balance Board

While, evidently, research attempting to quantify balance ability using force platforms was already conducted before the WBB's introduction in 2007, its release made working with force platforms widely accessible and popular due the WBB's low cost compared to laboratory-grade force platforms and a flourishing developer community [37]. The WBB's low cost and it's primary usage as gaming periphery raises the question if both systems produce data of comparable quality. The validity of the WBB as sensory device for assessment of postural control is subject of various research papers [6][38][37]. In their work, *Clark et al.* [6] compare COP measurements taken from laboratory-grade force platforms with those produced by a WBB. Their results suggest that the WBB is a valid tool for assessing standing balance, thus COP measurements are of comparable quality to the gold standard in clinical practice, laboratory-grade force platforms. *Holmes et al.* [38] come to the same conclusion comparing COP measurements of patients suffering from Parkinson's disease. Figure 2.1 compares 20 seconds of COP measurements taken from a WBB and a laboratory-grade force plate, model AMTI OR6-7-2000, produced in the US.



(a) Medio-lateral COP (COPx) (b) Anterior-posterior COP (COPy)

Figure 2.1: Comparing sensory data of WBB and laboratory-grade force plate [37].

As a legitimized alternative to laboratory-grade force platforms, the WBB opens up new opportunities for research in balance assessment and rehabilitation.

2.6 Wii Balance Board functionality and sensory data

The WBB was released by Nintendo in December of 2007 as additional input device for the Nintendo Wii gaming console. The Wii Fit gaming system aims to combine physical activity with the entertaining aspects of video games to ultimately make exercise more enjoyable. This concept peaked the interest of research in balance rehabilitation, where studies were conducted to evaluate the impact of off-the-shelf video games on balance rehabilitation. Utilizing the WBB and other peripherals of the Wii gaming system for research in rehabilitation is often referred to as *Wiisearch* [37] or *Wiihabilitation* [5]. Although playing off-the-shelf video games on the Wii can have positive effects on therapy success, the games are often too difficult for patients with mobility impairments [5]. The lack of customization, grading and measurement of progress resulted in efforts developing SGs for balance rehabilitation to more closely align with patient needs. Using peripherals of the Wii, such as the WBB, as input device for custom-made games requires an API that is capable of establishing a connection to the device and parse the data stream emitted from the device into a processable format.
The WBB, as shown in figure 2.2 **A**, consists of a 433 mm x 228 mm (L x W) rigid force platform connected to four uni-axial vertical force transducers installed in the foot pegs (figure 2.2, **B**). The foot pegs are located at the corners of the force platform, containing one force transducer each (figure 2.2, **C**). A force transducer is a load cell consisting of a metal bar with a strain gauge that converts applied force into a voltage [39]. The four force transducers of the WBB sample at a frequency of approximately 100Hz [39]. The WBB constantly emits a stream of data containing the measurements via Bluetooth to connected devices. A device paired with the WBB needs additional software to interpret and process the data stream received from the WBB. Due to the WBB's popularity in the *WiiHabilitation* community, a variety of APIs have been developed [40] that enable a stable connection and data exchange with the WBB.

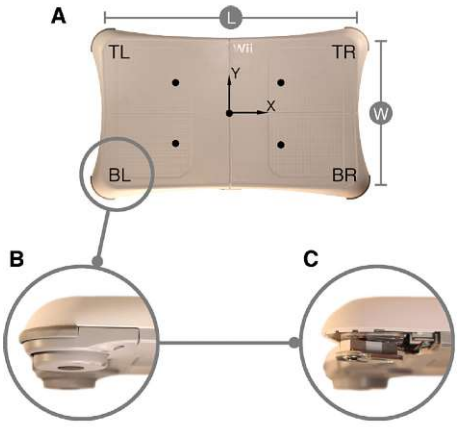


Figure 2.2: (A) Top-down view of the WBB, (B) Bottom-left (BL) foot peg, (C) Force sensor in foot peg [39].

2.6.1 Calculating center of pressure from WBB sensory data

The voltage measured at the four sensors of the WBB can be used to compute the center of pressure (COP), which is an approximation of the body's center of mass projected vertically onto the floor. Figure 2.2 **A** shows the coordinate system used to compute the COP in medio-lateral (ML) direction (x-axis, sideways movement) and anterior-posterior (AP) direction (y-axis, forwards-backwards movement and vice versa). The COP in ML direction, also referred to as COP_x or COP_{ml}, is the approximation of the center of mass projected onto the floor towards the left and right side of the body. When body sway increases from a central position to either side, e.g. through weight displacement from the left foot to the right foot, COP_x fluctuation increases. The COP in AP direction, also referred to as COP_y or COP_{ap}, meaning from front-to-back or back-to-front, increases when body sway towards the front or back is increased. COP is an important metric for assessment of balance ability because it approximates body sway [37]. It is calculated using the following equations [37][41][3]:

$$COP_x = \frac{L}{2} * \frac{(TR + BR) - (TL + BL)}{TL + TR + BL + BR} \tag{2.1}$$

$$COP_y = \frac{W}{2} * \frac{(TL + TR) - (BL + BR)}{TL + TR + BL + BR} \tag{2.2}$$

Equations for COP computation

In equations **2.1** and **2.2**, L refers to the length and W to the width of the force plate. TL, TR, BL and BR refer to the voltages or weights measured at the respective force transducer at the top-left (TL), top-right (TR), bottom-left (BL) and bottom-right (BR) of the force platform. The resulting COP values are within the intervals $[\frac{-L}{2}, \frac{L}{2}]$ for COP_x and $[\frac{-W}{2}, \frac{W}{2}]$ for COP_y. Each measurement constitutes a point in the coordinate system depicted in figure 2.2 **A**, associating COP_x with COP_y data over time. Visualizing the COP trajectories, combining COP_x and COP_y measurements, is referred to as a statokinesiogram. Figure 2.4 shows two examples for statokinesiograms, comparing COP trajectories from a WBB and a laboratory-grade force plate.

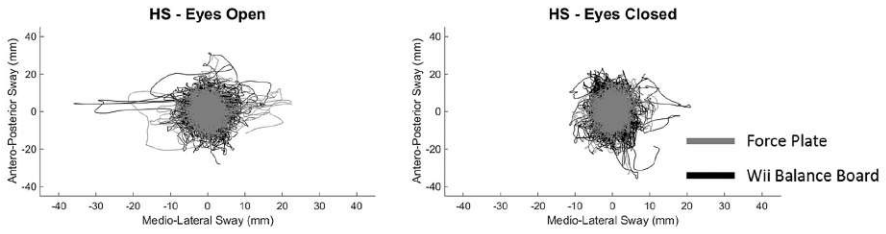


Figure 2.4: Statokinesiograms showing COP sway paths of healthy subjects (HS), comparing the WBB to a laboratory-grade force plate [41].

2.6.2 Quantifying balance ability

To generate levels based on player impairments, balance weaknesses must be identified. To achieve this, balance ability must be evaluated in a data-driven fashion. COP data constitutes the basis for most parameters that are utilized for quantification of balance ability. These parameters, when combined with learning algorithms, can be used to automatically detect if a person is at risk of falling [3] or approximate a patient's BBS [42]. In the context of this thesis, parameters to quantify balance ability are used to model balance behavior of healthy subjects, thus they constitute a vital ingredient for the development of the proposed prototype. Selecting the most viable COP-based parameters to quantify postural control is challenging because of conflicting opinions in literature [43]. In the following, the most widely used parameters derived from the COP are discussed in more detail.

Maximum and minimum amplitude A COP trajectory's maximum amplitude is the maximum absolute displacement of the COP from its mean, whereas the minimum amplitude is the minimum displacement from the mean. Increased maximum or minimum amplitude suggests a decreased ability to sustain an upright stance. A decrease in either of these values potentially refers to an increase in the ability to maintain balance. These parameters are 1-dimensional, thus can be calculated independently for AP and ML COP values. The maximum and minimum amplitudes are single data points, not incorporating any additional information from the COP trajectory, used to represent an entire trial of data. This make the maximum and minimum amplitude vulnerable to outliers, which increases variance between trials and subjects significantly [43]. For a WBB measuring roughly 100 measurements per second, a 30 second trial would result in approximately 3000 data points. For each of these data points, COP values are calculated. A single outlier amongst these 3000 COP measurements will lead to disappropriately high values for either maximum or minimum amplitude, preventing any reasonable deductions about postural stability or balance control.

Peak-to-peak amplitude The peak-to-peak amplitude is the difference between the maximum and minimum amplitude, thus it inherits the same weaknesses. High variability in maximum and minimum amplitudes directly translates into high variability in peak-to-peak amplitude. Maximum and minimum amplitude are singular values derived from the COP trajectory, hence are incapable to sufficiently capture measurements of a whole trial [43].

Mean amplitude The mean amplitude of a COP trajectory is an average value computed over all data points of a trial. It incorporates the whole data set and is a more reliable representation of postural control [43]. An increase in mean amplitude suggests a decrease in postural control, whereas a decrease is related to an increase in postural stability. Even though the mean COP is a more stable indicator for postural control than the minimum or maximum amplitude, it still is relatively sensitive to outliers.

Furthermore, COP-based parameters solely based on the x- and y-position of COP in the coordinate system, see figure 2.2, are sensitive to foot positioning on the force plate. Mean COP fluctuates with different degrees of stance width, stance length and foot angle. This weakness can be minimized by implementing an appropriate trial protocol and ensure correct foot positioning before each execution. Not considering this weakness results in considerable intrasubject variability and also increased variability between trials [43].

Sway path The sway path (SP), also referred to as COP path [6] or total excursion [43] constitutes the total length of the COP trajectory. Figure 2.5 visualizes the transformation of a COP trajectory into its sway path by imagining it as a wrinkled piece of string that is being pulled into a straight line. It is controversial to use the sway path as parameter to assess postural control because it is possible to see large values during a stable stance or small values during an unstable stance. *Palmieri et al.* [43] conclude that sway path is not useful to quantify changes in postural control.

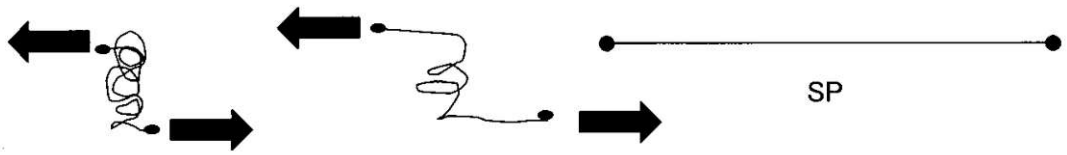


Figure 2.5: Transformation of COP trajectory to sway path [43].

The SP can be calculated by using equation 2.3, where N is the total number of COP measurements in the trial. The idea is to sum up all euclidean distances between measurements, where Δ_{ml} constitutes the difference between measurements in ML direction and Δ_{ap} refers to the difference between measurements in AP direction.

$$SP = \sum_{i=1}^{N-1} \sqrt{\Delta_{i_{ml}}^2 + \Delta_{i_{ap}}^2} \tag{2.3}$$

Equation for calculating the SP

COP velocity The COP velocity is calculated by dividing the sway path by the trial duration. *Palmieri et al.* found that "*an increase in COP velocity is thought to represent a decreased ability to control posture, whereas a decrease in the velocity represents an increase in the ability to maintain an upright stance*" [43]. When using either sway path or COP velocity, information about AP and ML movement is combined into a single metric, thus vital directional information is lost with these derivatives.

Sway area Sway area is calculated by multiplying the AP and ML peak-to-peak amplitudes. It can be visualized as a bounding box around the COP trajectory, containing all measurements. This metric suffers from the same drawbacks as minimum and maximum COP amplitudes.

Root-mean-square (RMS) amplitude The RMS amplitude corresponds to the standard deviation of COP displacement (see equation 2.4), which measures the average absolute displacement around the mean COP. A decrease in RMS amplitude is linked to an increased ability to preserve an upright stance, whereas an increased value suggests a decreased ability in postural control [43].

$$RMS_{amp} = \sqrt{\frac{1}{N} * \sum_{i=1}^{N} (x_i - \mu)^2} \tag{2.4}$$

Equation for calculating the RMS amplitude

Root-mean-square (RMS) velocity The RMS velocity is defined as the "*distribution of COP displacements over time*" [43]. Heightened RMS velocity corresponds to increased COP displacements, which suggests hasty or uncontrolled weight distribution during trial and therefore a weakness in postural control. Similarly to RMS amplitude, a decrease in RMS velocity represents increased ability to maintain an upright stance. Both RMS amplitude and RMS velocity have been suggested to be reliable measures to evaluate postural equilibrium [43], which refers to *coordination of movement strategies to stabilize the center of body mass during both self-initiated and externally triggered disturbances of stability* [44]. Furthermore, both RMS amplitude and RMS velocity show sufficient intrasubject consistency reliability across multiple sessions [43].

$$RMS_{vel} = \sqrt{\frac{1}{N-1} * \sum_{i=1}^{N-1} (\Delta_{i,i+1} - \bar{\Delta})^2} \tag{2.5}$$

where

$$\Delta_{i,i+1} = |x_{i+1} - x_i| \tag{2.6}$$

and

$$\bar{\Delta} = \frac{1}{N-1} * \sum_{i=1}^{N-1} \Delta_{i,i+1} \tag{2.7}$$

Equation for calculating the RMS velocity

A velocity is calculated by setting a distance in relation to the time elapsed traveling said distance. Assuming a constant sampling rate, e.g. 100Hz for the WBB, the time dimension for velocity calculation is inherent. In other words, at a constant sampling rate, the

number of measurements taken reflects the time elapsed during trial. If time can be seen as constant, which is the case if measurements are taken every 1/100th of a second, the only variable to consider becomes distance. The elapsed time can be substituted with the number of absolute distances between measurements taken into account. Consequentially, the RMS velocity can be defined measuring the average absolute displacement around the mean displacement. $\Delta_{i,i+1}$, see equation 2.6, refers to the absolute displacement between two consecutive COP measurements, where $\bar{\Delta}$, see equation 2.7, constitutes the mean absolute displacement. The RMS velocity can be calculated in both ML and AP direction, which results in horizontal and vertical COP velocities. The ability to investigate ML and AP velocities independently, similarly to the RMS amplitude, makes these metrics highly valuable for quantification of postural control [43].

Evidently, a lot of information about a person's postural control can be derived solely based on COP data. Since the WBB represents a valid low-cost alternative to laboratory-grade force platforms, all of the described COP-based metrics for quantification of balance ability can be calculated utilizing data from the WBB with sufficient accuracy. This additional information about postural control and balance ability of patients can also be used to personalize experiences in game environments. It is shown that SGs can have a positive impact on therapy success. Developing SGs to better accommodate patient needs instead of using off-the-shelf video games is an important step, but personalization of therapy does not have to stop there. Tailoring the experience of a SG to better adjust to patient impairments, utilizing data gathered from previous trials to personalize in-game experiences, could further motivate and engage patients on their road to recovery. Machine learning techniques have proven to be useful in many areas where raw data is turned into valuable insight [45]. This insight can also benefit patients in balance rehabilitation.

2.7 Machine learning with WBB sensory data

In their work, *El et al.* [45] describe machine learning as "*an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment*". Machine learning techniques have been applied to various fields, such as pattern recognition, computer vision, finance, computational biology and medical applications, with great success [45].
Machine learning algorithms can be broadly categorized in supervised, unsupervised and reinforcement learning algorithms [46][47].

2.7.1 Categorization of machine learning algorithms

Supervised learning A supervised learning algorithm utilizes a labeled dataset for training. Training refers to the generalization of input data to predict the outcome for a new data point. It is called *supervised*, because the algorithm utilizes the association

between labels and datapoints to learn. Annotation of training data is performed by domain experts, thus it learns in a *supervised* manner [46].

Unsupervised learning Unsupervised learning does not rely on labeled training data. It uses techniques to organize data based on hidden patterns found in underlying data, e.g. through clustering. Subsequently these clusters can be used to distinguish between different classes of data and classify new data points [46].

Reinforcement learning Whereas in supervised and unsupervised learning algorithms, training is performed as a separate step before new data can be inferred, in reinforcement learning the training and testing phases overlap. New data continuously triggers mechanisms to reiterate on learned behavior, contributing to the learning process [46].

2.7.2 The machine learning pipeline

Even though there are major differences between types of machine learning algorithms, the fundamental flow from data ingestion to model deployment and feedback, also called the *model lifecycle* (figure 2.6), is similarly structured. The so-called *machine learning pipeline* implements the steps defined in the *model lifecycle* and aims to modularize the required stages to ultimately automate all of its steps [48].

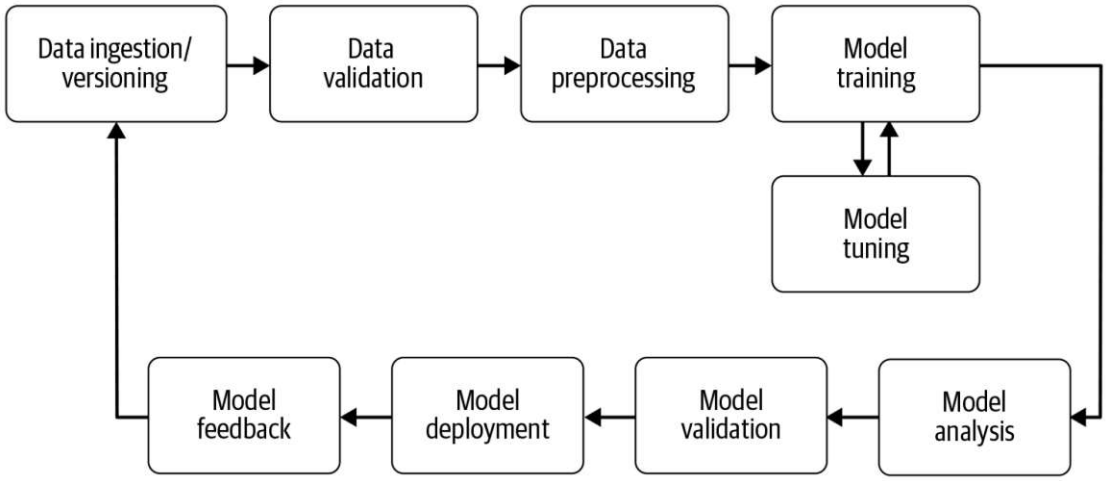


Figure 2.6: The model lifecycle [48]

Data acquisition This step is not mentioned in the model lifecycle, because it is performed before the first step and can not be automated. Quality of data is highly dependent on this phase, since it could either stem from controlled experiments that followed a routine protocol or a noisy stream of unstructured data. In section 2.6.2 the impact of foot positioning on the force plate affecting calculation of the mean COP

is discussed. Not consistently following the same instructions during a trial results in insufficient data quality and ultimately poor performance of the resulting model.

Data ingestion In the first step, depicted in the model lifecycle, data is transformed into a processable format that can be digested by the next steps in the pipeline. Figure 2.6 also mentions versioning as part of this step. Performing versioning is good practice at this step, because it allows reproducing the trained model with the current snapshot of data [48]. Due to the rapid prototyping approach in this thesis, which does not foresee a continuous evolution of the underlying model, versioning is not considered.

Data validation After the data is in an accessible format, statistical outliers have to be detected and removed from the dataset. If data is used for a classification task, investigation for imbalanced data should be part of this step. In a labeled dataset, no class should be significantly over-represented to prevent introducing bias to the model. This step can be automated by introducing validation metrics and mechanisms that stop the pipeline and trigger alerts when predefined criteria are not met. Furthermore, feature engineering is performed as part of this step [48]. In a classification task, features are the properties that best distinguish between the classes. Feature engineering is the process of finding the combination of properties (features) that maximize the probability of correctly classifying new data points.

Data preprocessing In this step data is converted into a format that can be used to train the model. The numerical values of features can vary in range. The process of bringing feature values into the same numerical range is referred to as feature normalization [48].

Model training & tuning The model training constitutes the core of the machine learning pipeline. In this step, the model is trained, based on the features constructed in the steps prior. In supervised algorithms, training refers to learning and creating a generalization of the training data. In unsupervised learning, when using a large amount of data, training can require a significant amount of time and resources, thus tuning the training process has gained more attention. Investigating the optimal combinations of parameters to use for training a machine learning model can significantly improve performance [48].

Model analysis & validation In this step, accuracy of predictions made by the model are investigated. Researching the impact of features on the result is also part of model analysis [48].

Model deployment After performance of the model is validated the model can be deployed to be accessible from other services [48].

Model feedback In the last step of the pipeline, valuable insight about performance and effectiveness of the deployed model is gathered. New data can be captured to improve the model over time and make it more robust [48].

In this thesis, the machine learning pipeline is followed for modeling healthy balance ability. Section 4.6.2 shows how development of the underlying model for quantifying balance ability leverages the structure described by the lifecycle model, displayed in figure 2.6. An unsupervised learning approach is utilized to build the model for this thesis, which is solely trained on data gathered from healthy individuals performing weight shifting exercises on the WBB. The following section outlines related state-of-the-art work for quantifying balance ability and describes how unsupervised learning techniques can be utilized to learn from one-sided data.

2.7.3 Learning from unbalanced datasets

Machine learning algorithms in balance rehabilitation are primarily used to estimate balance ability [9][42] or the risk of falling [3]. To verify the accuracy of a model that estimates balance ability, a frame of reference is required. In their work, *Bacciu et al.* [9] collected data using a WBB, where the acquired data of each subject is associated with their BBS. Similarly, *Bao et al.* [42] label their training data with a balance score of 1-5, evaluated by a physiotherapist. Associating the training data with a frame of reference, in this cases the BBS or a custom scoring metric, is referred to as labeling the training data. Hence, both these approaches use supervised learning algorithms. The process of labeling is resource-intensive since it requires time and dedication of a professional in the respective field.
For estimating if a person is at risk for future falls, sufficient data of both fallers and non-fallers has to be aquired [3]. Estimating an accurate balance metric based on force plate sensory data requires trial data of over 20 individuals with balance impairments and also pre-trial evaluations of balance ability by physiotherapists [9]. At the foundation of every machine learning approach is data acquisition and healthcare-related data is especially hard to obtain.
To overcome this issue, the approach used in this thesis is to quantify the deviation of a patient's balance ability from a healthy baseline. This allows sampling data from non-impaired people, attempting to model healthy balance behavior and compute the deviation from the healthy baseline to quantify balance ability. To achieve this, a machine learning technique referred to as *anomaly detection* is utilized. This class of algorithms belongs to unsupervised learning techniques and is especially useful when working with *skewed datasets*. Skewed datasets refer to data that overwhelmingly contains examples of one particular class. Anomaly detection solves binary classification problems, as it can only differentiate between an anomaly (or outlier) and inliers (or non-outliers). In order to achieve this, any anomaly detection algorithm needs at least one data point that constitutes an outlier to establish a decision boundary. The decision boundary is a function that acts as a threshold above which a data point is considered an outlier, see figure 2.7. The 2D data points (or *observations*), represented by *Feature 1* and *Feature 2*,

are normally distributed. The purple ellipsis designates the decision boundary. Given the distribution and decision boundary of the 2D Gaussian distribution shown in figure 2.7, an observation is considered an inlier if $p(X) \geq \epsilon$, where $p(X)$ is the probability, defined as $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}(\frac{x-\sigma}{\sigma})^2}$, and ϵ the threshold defined by the decision boundary. If the probability for an observation belonging to the underlying distribution is below the threshold ϵ, or in other words, if it is unlikely that the observation belongs to the class modeled by the Gaussian distribution, it is considered an outlier.

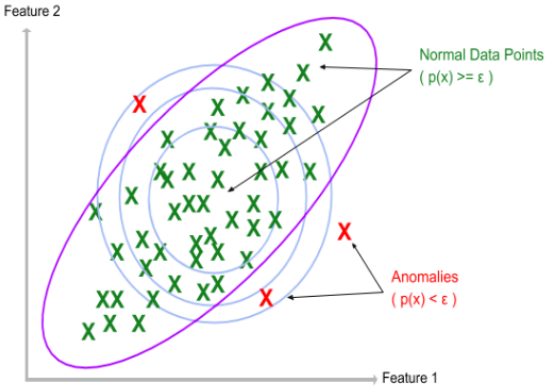


Figure 2.7: 2D Gaussian distribution with decision boundary [49]

Within the realm of anomaly detection algorithms, there exists a special variant of classification algorithms, the one-class classifiers (or *unary classifiers*, also referred to as *class-modeling*). This family of classifiers only needs to be trained on one class, also referred to as *target class* [50]. This opens up the possibility to model a specific class, e.g. balance ability of healthy individuals, without the need of outlier data. Thus, decision boundaries can be derived only considering data from the target class.
In the context of this thesis, this means that *class modeling* can be used to learn healthy balance behavior by only utilizing data from healthy subjects and calculate the deviation from healthy balance to quantify balance ability. Furthermore, after modeling healthy balance ability, a threshold, or decision boundary, can be determined to distinguish between healthy and impaired balance ability. The achieved balance scores for each level can be used to subsequently parameterize level generation.

2.7.4 Personalization of game difficulty

Personalization of the game environment and adaptation of game difficulty has shown to improve patient motivation and engagement [7]. In their work, *Borghese et al.* [8] state that SG in balance rehabilitation have to be parametric to adapt to patient impairments.
"*Specifically, games should be defined by parameters that can be regulated and adapted, depending on the level of game difficulty. Such parameters are initialized inside the*

hospital, where the clinician prescribes the therapy and are continuously adapted to each patient's progression." [8] Instead of increasing game difficulty based on the number of in-game objectives reached, as demonstrated in the work of *Gil-Gomez et al.* [7], the output of a machine learning model that aims to describe the actual balance performance can be utilized to adapt to player impairments. In the context of balance rehabilitation, COP-based parameters can be derived from WBB sensory data that contain viable information about patient balance ability.
The prototype developed in this thesis utilizes this data to detect deficiencies in balance ability and based on the insight gained, changes level layout and size to accommodate the patient's therapy needs. To achieve this, in-game adaptation of the game level is discussed with physiotherapists to ensure appropriate adaptation of level difficulty based on patient performance. This is directly related to the second research objective of this thesis. To develop a prototype that is of value for therapists and patients with balance impairments, requirements for a system that can support the rehabilitation efforts have to be gathered. The following section describes the process of requirements engineering and techniques utilized to collect the information necessary for developing an adaptive SG for balance rehabilitation, generating personalized levels based on data-centric evaluation of balance ability.

2.8 Requirements engineering

The requirements engineering process is a vital part of every software development project. "*The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended. Broadly speaking, software systems requirements engineering is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation*" [51]. Given this definition, the first step of successful requirements engineering is to identify stakeholders of the software and subsequently gather information about their needs, also referred to as requirements elicitation. Stakeholders are defined as persons or organizations who "*have an active interest in the system because they'll actually use it or are directly involved in processes that the system will change*" [52].
The requirements engineering process can be subdivided into requirements elicitation, analysis, specification and validation [53], see figure 2.8. The requirements elicitation phase is regarded as the most crucial phase in requirements engineering [54], because the obtained requirements should sufficiently reflect expectations of various stakeholders and draw an outline of what is expected of the resulting system. In the analysis phase, the gathered requirements are reviewed to reinforce mutual understanding and negotiated to evaluate which requirements are most important or can be left out entirely based on the evolved understanding of the system. The analysis is followed by a specification phase, which has the purpose of formalizing the requirements in an unambiguous way and provide an accessible and understandable documentation for all stakeholders. As a last step, the formalized requirements are validated against the stakeholders' interests to verify that they accurately describe what the system needs to achieve.

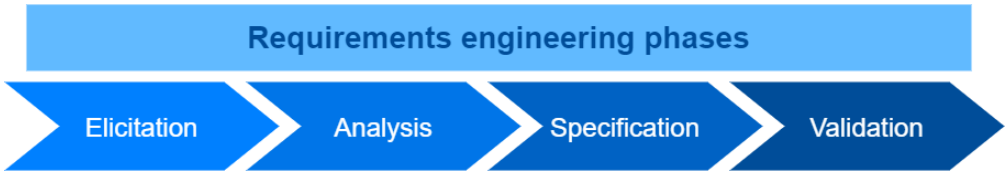


Figure 2.8: Common phases for requirements engineering

Requirements can be classified as functional and non-functional.

Functional requirements In software engineering, a functional requirement specifies a function that the software system must be capable of performing. In other words, they define the fundamental process or transformations that software components perform on inputs to produce desired outputs. Functional requirements define the behaviour of a software system [55].

Non-functional requirements As opposed to functional requirements, non-functional requirements "*do not describe what the software system will do, but how the software will do it*" [55] through performance, design or quality requirements.

As the requirements elicitation is deemed the most influential part of requirements engineering, the following section provides insight into common techniques utilized during this phase.

Requirements elicitation

Requirements elicitation refers to the practice of discovering the requirements of a system from users, customers or other stakeholders. The quality of the system is highly dependent on the requirements defined during this phase [54]. Requirements elicitation techniques are the ways and procedures to obtain user requirements. Common examples are brainstorming, interviews and questionnaires. The resulting requirements are used to develop the system to ultimately satisfy the needs of stakeholders. Requirement elicitation techniques can be roughly categorized in qualitative and quantitative methods. Qualitative methods revolve around gathering in-depth information from a small group of people and subsequently interpret the results. This can be done by brainstorming or unstructured interview, which leaves a lot of space for open discussion and generation of new ideas. Qualitative methods aim to yield highly detailed insight about specific topics. On the other hand, quantitative methods focus on collecting information from a variety of sources, e.g. by a structured interview or literature research, and statistically evaluating the results. The following paragraphs describe the most common qualitative and quantitative requirements elicitation techniques, based on the work of *Yousuf et al.* [56], in more detail.

Document analysis This quantitative elicitation technique revolves around collecting and analyzing data from existing literature to gain extensive insight into a specific topic. Document analysis is helpful either when stakeholders are not available, to complement information gained from other techniques or to gain insight into a topic before applying other techniques. [56]

Brainstorming This technique describes sharing of ideas in an unstructured and conversational way with one or more stakeholders. Brainstorming focuses on a particular issue and encourages innovative ideas by creating an unconstrained space for discussion, thus it constitutes a qualitative method for requirements elicitation. The more competent the participants of the brainstorming session are regarding the focused topic, the higher the quality of acquired data will be. [56]

Interviews Interviews are face-to-face conversations with stakeholders asking questions and documenting the results which finally produces requirements. Based on preparation of the questions and the degree of guidance during the conversation, interviews can be categorized as structured (closed), semi-structured or unstructured (open) interviews. Structured interviews are formal and consist of a set of predefined questions. In semi-structured interviews a combination of predefined and unplanned questions are asked. Both of these techniques provide quantitative data, because constraining the discourse about a topic does not create sufficient space for open conversation and discussion of new ideas. Complementing these techniques is the unstructured interview, where unplanned questions are asked in an informal fashion that provides qualitative data. Unstructured interviews create a deep understanding of an issue but are also more time-consuming than structured or semi-structured interviews. [56]

Questionnaires A questionnaires (or survey) is a quantitative method used when face-to-face conversation with stakeholders is not possible. They contain clear, well defined and precise questions and are usually meant to collect requirements from a large group of stakeholders. The questionnaire is considered the cheapest way of requirements elicitation, because it can reach a large number of people in a short time in an economical fashion. A disadvantages is that they can be misinterpreted if questions are ambiguous. [56]

Prototyping With this technique, a prototype of the desired system is produced in an iterative fashion by improving the system step-by-step based on stakeholder feedback. It is especially useful when developing new systems. Each consecutive version of the prototype communicates ideas to the stakeholder. These ideas can be discussed and iterated on by incorporating feedback into the development process, thus ultimately creating a system that achieves stakeholder satisfaction. [56]

The contributions of this thesis heavily rely on understanding and evolving the current state of the art around SGs in balance rehabilitation, which are discussed in the following chapter.

CHAPTER 3

State of the art

This chapter starts with giving insight into practical examples of SGs in healthcare for the application areas discussed in section 2.2. This shows how the topic of this thesis is embedded into the landscape of SGs in healthcare. Subsequently, the state of SGs for balance rehabilitation utilizing the WBB is briefly described, followed by proposed solutions to evaluate balance ability based on sensory data sampled from the WBB. Finally, the use of adaptation mechanisms to adjust game difficulty is described and how the contributions of this thesis enhance the current state of the art.

3.1 Serious Games in healthcare

Given the rapidly growing landscape of SGs in healthcare, with a vast variety of application areas and use-cases, the need for a taxonomy arose. *Sawyer et al.* [21] propose such a taxonomy for SGs in the healthcare sector, defining prevention, therapy, assessment and education as the main categories. Section 2.2 contains definitions for each of these classifications and provides further insight regarding taxonomy selection. In the following, state-of-the-art research for each of these category is discussed.

3.1.1 Preventative Serious Games

Young et al. developed a SG [57] for elderly people with the goal of increasing their balance ability and self-efficacy with sensory-motor training. Balance impairments lead to an increased risk of falling and become more prevalent above the age of 65 [3]. Fall incidents not only lead to physiological injuries but also impact on the psychological well-being of the person that fell, thus preventative measures are essential. *Young et al.* utilized a WBB as sensory device, measuring the center of pressure on the force plate to control the patient's position in the level. The balance exercise performed during gameplay evidently increased the patient's balance skill and furthermore, the patients were motivated to continue their training efforts. Further preventative SG focus on prevention of smoking [58],

asthma prevention [59], HIV prevention [60], prevention of violence [61] and nutrition [62].

3.1.2 Serious Games for therapy

Ma et Bechkoum developed a series of virtual reality SGs to encourage patients' physical activity in motivating virtual environments that adapt based on in-game performance [22]. The games are targeted at stroke patients and aim to support them regaining natural movement patterns through virtual movement therapy. One of the games is called catch-the-orange, see figure 3.1.

(a) In-game virtual environment [22]

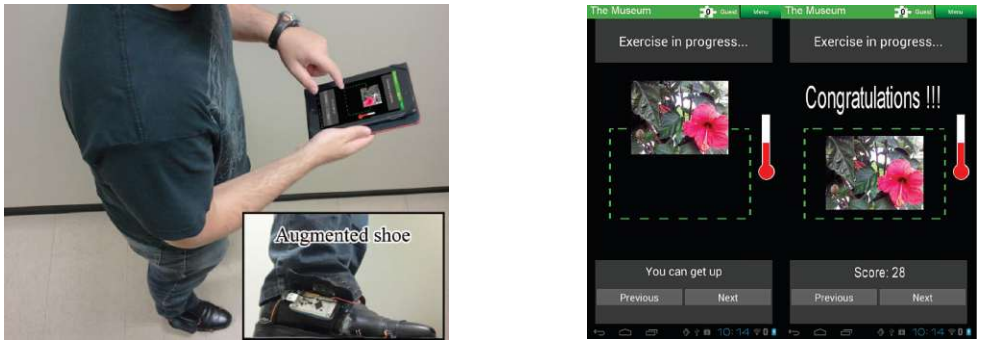
(b) A person playing the game [22]

Figure 3.1: Catch-the-orange game developed by *Ma et Bechkoum* [22]

In this SG, the player holds a virtual basket to catch oranges falling from random directions onto a target area. Position and orientation of the in-game basket are controlled by sensors on a real basket in the player's hands. To adjust to patient performance, variations in basket and orange size, the falling speed and time between oranges falling were employed. Their results show that SG intervention had a positive impact on recovery and the patients reported enjoying the games [22].
Caglio et al. [23] propose a SG for cognitive rehabilitation after traumatic brain injury. Patients suffering from traumatic brain injury often have problems regarding memory, attention and executive functioning, the most common issue representing memory problems. They developed a 3D virtual reality SG for navigational training to investigate its effect on both spatial and verbal memory on a 24-year-old suffering from traumatic brain injury. They concluded that "*virtual navigational training might support an enhancement of memory functions in brain-damaged adults*" [23]. Their research also suggests that training in a virtual environment can transfer to improve real world performance regarding spacial and verbal memory.

3.1.3 Serious Games for assessment

Brassard et al. [32] developed a SG to assess standing balance. Within the game, the player has to navigate through a museum, collecting puzzle pieces in the process. To collect puzzle pieces, the player has to perform exercises from the Berg Balance Scale (BBS). Figure 3.2b depicts the user interface of an exercise. To measure the player's performance, they have to wear a shoe equipped with a variety of sensors, see figure 3.2a. The data gathered during the assessment by the sensory device is evaluated and a score according to the BBS is calculated.



(a) Performing exercise with sensory-enhanced shoe [32]

(b) Execution and conclusion screen of an exercise [32]

Figure 3.2: Balance assessent game developed by *Brassard et al.* [32]

Brassard et al. conclude that their system is effective in evaluating the patient's balance ability. SGs can be utilized to enhance patient motivation and engagement in the context of assessing medical conditions.

3.1.4 Serious Games for education

In their work, *Ricciardi et al.* [24] found that SGs "*represents a useful training technology in health professions and should be considered as an effective training tool.*" Furthermore, they found that SGs improved learning skills acquisition in some cases.
Parvati et al. [63] designed the SG *CliniSpace* to expose residents and nurses to training experiences involving many types of emergency patients in an immersive 3D multiplayer environment, see figure 3.3.

Figure 3.3: View of emergency room showing interactive objects [63]

Due to the online multiplayer functionality of the game, the learning environment can be scaled up according to class size. Immersive online medical environments, as provided in *CliniSpace*, have been shown to be effective for scenario-based learning [63].
Qui et al. [64] designed the SG *Stop the Fountain* with the purpose of training surgeons on blood management in mind. Blood management is crucial in surgical procedures, because blood loss is a major cause of morbidity in many traumatic situations. The game provides a 3D virtual environment and can be played in training mode, which incorporates hints into the gameplay to guide trainees through the procedure. Furthermore, there is a time-attack mode that focuses on fast and correct execution of the procedure and a collaborative mode in which several trainees can complete a task together. Figure 3.4 depicts a training session of the game. The game screen includes vital signals, blood volume, remaining time and available medical instruments or interventions to choose in order to interact with the patient and ultimately stop the bleeding.

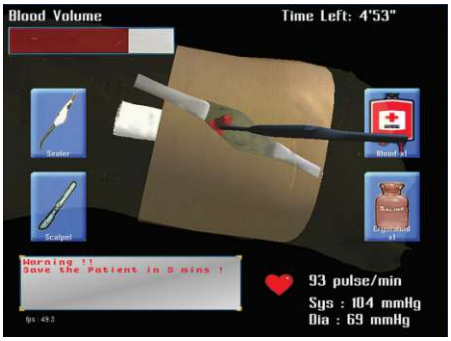


Figure 3.4: Training session which requires sealing the blood vessel [64]

The authors of *Stop the Fountain* found that the game-based training environment can improve learning of the psycho-motor skills used in blood management [64]. Trainees who pre-trained without the games reached a lower maximum score.

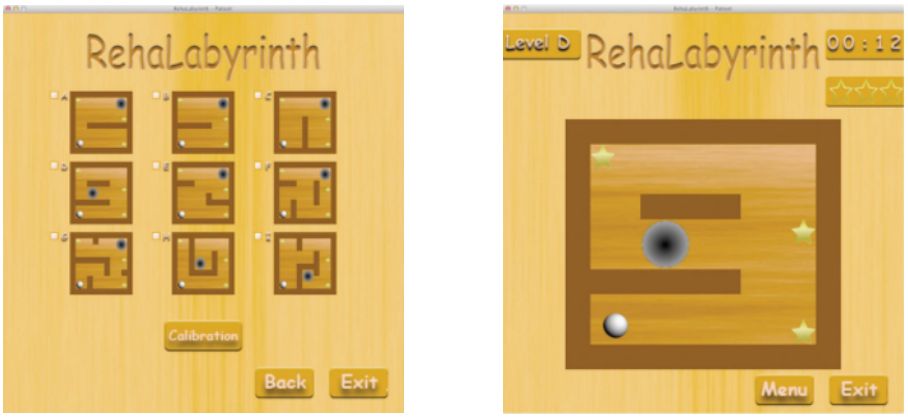
Given the above taxonomy for SGs in healthcare, the work in this thesis focuses on the rehabilitation aspects by designing and developing a new exergame that leverages personalized level generation to improve patient motivation.

3.2 Serious Games for balance rehabilitation utilizing the Wii Balance Board

As mentioned in chapter 2.4, the WBB drastically accelerated research in SG for balance rehabilitation using force plates due to its low price and high sensory accuracy, even when compared to laboratory-grade force plates. The following sections describe state-of-the-art research for SGs (and systems that utilize SGs) in balance rehabilitation in combination with the WBB.

3.2.1 RehaLabyrinth

RehaLabyrinth [33], developed by *Baranyi et al.*, is a SG for post-stroke balance rehabilitation with the aim to boost motivation and engagement during rehabilitation efforts. Figure 3.5a exemplifies a level overview with 9 unique mazes for a game session. To complete a session, each of these mazes has to be solved individually by navigating the white ball, shown in figure 3.5b, into the black circle by performing weight shifts on a WBB. The game incorporates competitive elements with a highscore list ranking player performances. Performance is computed as a combination of the time taken to solve the maze and the stars collected along the way.



(a) Level overview of *RehaLabyrinth* [33] (b) Gameplay of *RehaLabyrinth* [33]

Figure 3.5: *RehaLabyrinth* [33]

Furthermore, figure 3.5a shows the option to perform a calibration, which adjust sensitivity of WBB input to allow players with severe one-sided balance impairments to interact with

the game. After completing a session, objective data is provided through a web-interface, summarizing results, such as time taken to solve the mazes and achieved scores, and providing a visualization of the COP sway path for each maze. To support patients with balance constraints or adjust to a specific weakness in weight displacement, predefined levels can be adjusted through a level editor.
In case of *RehaLabyrinth*, adjusting level design to more precisely encompass patient impairments is done manually by the physiotherapist. The aim of this thesis is to develop a novel SG, inspired by the simplicity in game design and interaction mechanisms of *RehaLabyrinth*, which automatically generates game levels that encompass patient weaknesses detected during gameplay.

3.2.2 SilverBalance

SilverBalance [65] is an Exergame that uses the WBB as input device. The main game mechanic is to prevent collisions with in-game obstacles by performing weight shift exercises. Figure 3.6 shows the player, represented by a red square and an obstacle as a black bar coming from the top, moving slowly towards the player. Length and positioning (left or right) of the obstacles are randomized. This constitutes the first of two tasks implemented in *SilverBalance*. Difficulty is gradually adjusted over time by speeding up progression of obstacles towards the player. The second task involves vertical weight shifts. Players are confronted with obstacles taking up the whole width of the game screen and they have to jump over the obstacle by shifting their weight into the upper half of the WBB [65].

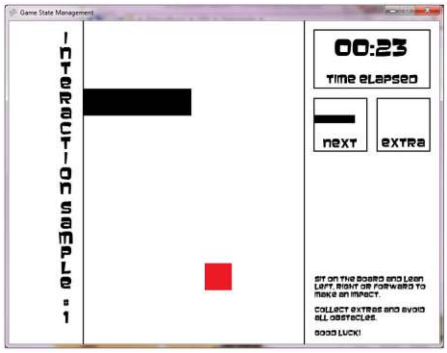


Figure 3.6: Graphical design of SilverBalance [65]

Evaluation of *SilverBalance* with nine senior citizens, aged 77 to 91, found that the simplistic game design was beneficial because it allowed them to focus on the game more easily. Furthermore, they found that even though the game's simple design, the group engaged in competitive play by commenting on each others results and comparing their performance [65].

Gerling et al.'s findings suggest using a simple design when developing SGs for elderly patients is beneficial. One shortcoming of *SilverBalance* is that the second exercise only relies on weight shifts towards the top half of the WBB. It is missing an exercise that allows for training anterior-posterior weight shifts towards the bottom half of the WBB. Various other SGs offer game mechanics that allow for full medio-lateral and anterior-posterior weight shifts during gameplay [33][66]. Furthermore, *SilverBalance* does not utilize the data gathered during trials. Game difficulty changes over time, ignoring previously conducted exercise sessions or progress during the current session. This data could be used to identify weaknesses in the patient's balance ability and incentivize weight shifts towards the weaker directions by spawning more obstacle at the appropriate positions.

3.2.3 eBaViR

eBaViR [7] is a gaming system that aims to improve standing balance in patients with acquired brain injury (ABI). ABI is the main cause of death and disability in young adults and in most cases, survivors experience balance problems, resulting in functional impairments that negatively impact their quality of life. *Gomez-Gil et al.* [7] propose *eBaViR* with three core goals in mind: (1) create an adaptive balance rehabilitation system for patients, (2) achieve a system that reinforces motivation and engagement and (3) provide therapists with objective data for evaluation.
To achieve their first goal, *eBaViR* offers the ability to manually customize gameplay elements, such as size or number of game elements, before starting a session, as shown in step (1) of figure 3.7. Furthermore, before each session, a calibration is performed, see figure 3.7 step (2), that measures the maximum excursion a patient is able to achieve in each direction. This is used to adapt the range of motion necessary to navigate the game levels, thus allows the system to tailor gameplay towards the individual needs of patients. The second target is reached by offering three SGs that use the WBB as input device. Interaction with the in-game environment is done by performing weight shifts. Similarly to *SilverBalance* [65], the games are designed in a simplified fashion to allow patients with cognitive impairments to focus on the exercises. Figure 3.7 shows a static image of gameplay in step (3) and also a pause screen in step (4). The number of pauses is defined in the pre-game configuration phase.
Lastly, they provide the calibration data and the scores reached after each session to the therapist in accordance with their third goal [7].

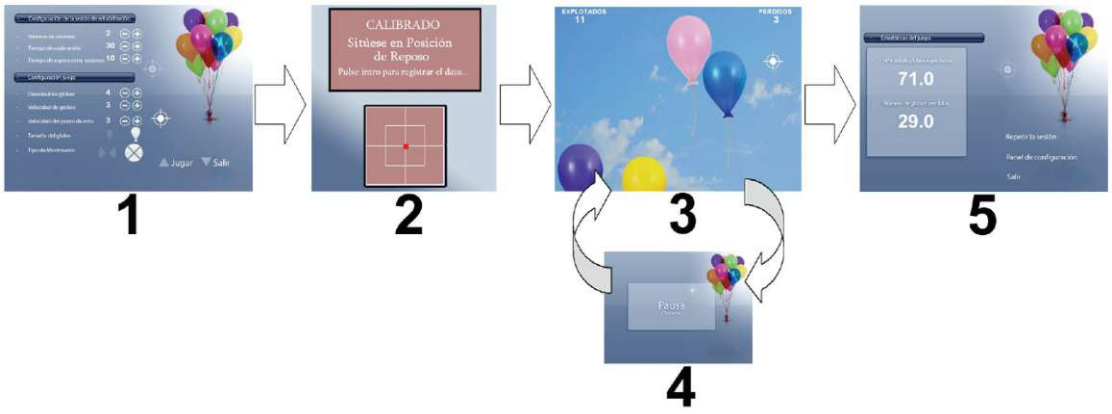


Figure 3.7: The flow of *eBaViR* is divided into 1. Setup, 2. Calibration, 3. Gameplay, 4. Break and 5. Scores [7]

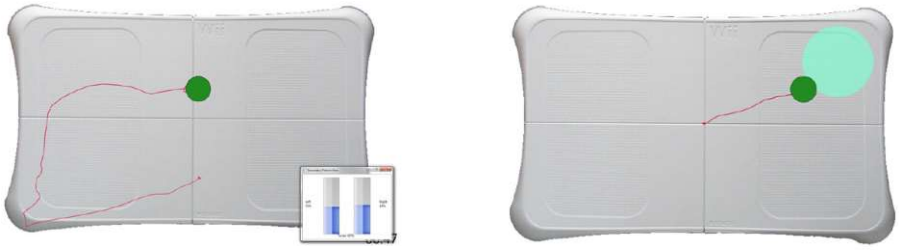
Gomez-Gil et al. developed *eBaViR* [7] with adaptive gameplay as a core aspect of their system. The configuration of in-game elements and pauses, as well as the calibration need close supervision of a therapist. Although they never claim that *eBaViR* is a at-home rehabilitation system, this is seemingly the only argument against it, since the WBB is a portable low-cost training device. Furthermore, their third goal is to provide therapists with objective data about patient performance. This data basically consists of the pre-game configuration and an amount of points reached during exercise, which not directly correspond to balance-specific performance. Using the balloon game, depicted in figure 3.7 step (3) as example, maximizing the speed and number of balloons, a patient could reach a much higher score by sheer coincidence (balloons spawning exactly at the patient's player position). Thus, the score at the end of a level should not be considered as an indicator of balance ability and should also not be taken into consideration when investigating balance improvement.

Data-based insight into a patient's balance ability should only be gathered from the actual source data, which in the case of *eBaViR* is the WBB's sensory data stream.
The SG developed in this thesis uses COP-based parameters for balance estimation, directly derived from COP measurements during gameplay. Furthermore, adaptions made to the game environment, more specifically, the level design, are derived from historic and intra-session balance performance. Alongside a final score as measurement for balance ability, the parameters computed for each level are presented after completion.

3.2.4 WeHab

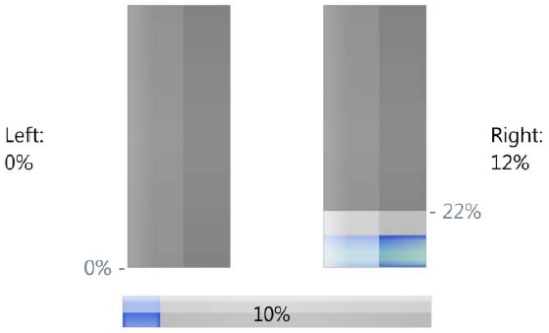
The *WeHab* system, proposed by *Kennedy et al.* [1], enhances balance rehabilitation by using visual feedback and providing additional insight by calculating COP-based parameters from WBB sensory data. Visual feedback during balance rehabilitation

sessions is achieved by tracing the COP on a computer screen while performing different exercises on a WBB. Figure 3.8 examplifies COP trajectories produced by the three exercises evaluated using *WeHab* system.
During the sit-to-stand transition, the patient starts in a sitting position with their feet on the WBB and attempts to stand up without external support. Figure 3.8a shows the subject's preference of placing weight primarily on their left leg during this exercise. A more optimal result for this exercise is a trace centered around the vertical axis with minimal lateral excursion. The bars on the bottom-right of figure 3.8a show the weight distribution between the left and the right leg at the final standing position.
Figure 3.8b shows the second exercise offered by the *WeHab* system, which is weight shifting. The larger light-green circle marks the target region, and the smaller dark-green circle references the patient's COP. The red line represents the patient shifting towards the target area. The patient is supposed to reside in the target area for 5 seconds. After the time has elapsed, the target region changes positions.
During the stepping exercise, shown in figure 3.8c, the patient places one foot on the floor in front of the WBB and taps onto the WBB with the other foot. In the example shown in figure 3.8c, the patient taps with their right foot with 12% of their body weight. The maximum value they should tap with is 22% of their body weight. The horizontal bar shows the total body weight that was put on the board, rounded to the nearest 5%.



(a) Display during sit-to-stand transition [1].

(b) Display during weight shifting [1].



(c) Display during stepping [1].

Figure 3.8: The *WeHab* system [1]

Even tough previous research has not shown clear evidence, that visual feedback enhances training success, *Kennedy et al.* found that therapists appreciated the ability to receive quantitative data during rehabilitation sessions instead of relying on visual estimations [1].
During every rehabilitation session using the *WeHab* system, COP data was collected. Utilizing these measurements, COP-based parameters, such as the mean COP velocity (COP velocity) and the standard deviation of COP displacement (RMS amplitude), were calculated. After all trials concluded, these metrics were used to calculate a score for each rehabilitation session. These scores were compared to balance ability estimations performed by therapists. This comparison showed no significant correlation between the calculated score and the balance ability estimation by the therapists. Since *Kennedy et al.* do not disclose how the final score is calculated, it is impossible to reason about this aspect of their work.
The *WeHab* system is designed solely for use in rehabilitation sessions under close supervision of a therapist. Furthermore, the exercises offered by the *WeHab* system were not designed as SGs, hence gamification aspects that could potentially further boost patient motivation are missed. Also, *WeHab* does not offer any mechanisms to adapt difficulty based on a patient's skill level. The quantitative data collected during trials was only used to gather further information during trials and calculate a score for each trial after trials have already concluded.
The prototype developed in the scope of this thesis is a SG that provides quantitative feedback after each level, enabling the patient to improve their score even within sessions and leverage gamification mechanics to enhance patient engagement and motivation. This quantitative feedback of the proposed prototype contains detailed information about the COP-based parameters, the deviation from the expected (optimal) parameters and a total score for the level. Furthermore, historic data about patient's performance, as well as current session data is used to personalize levels.

3.3 From Wii Balance Board sensory data to insight

The state of the art for SGs in balance rehabilitation lacks automated adaptation mechanisms based on sensory data. *eBaViR* [7] provides customization of game elements but does not utilize data produced during gameplay. *WeHab* [1] collects COP data and derives indicators for balance ability but only for means of computing a score and additional insight for therapists. They do not adapt their exercises based on the collected data.
To achieve meaningful personalization based on gathered COP measurements, balance metrics have to be chosen to indicate weaknesses in balance ability. Thus, this section discusses state-of-the-art research to quantify balance ability by investigating automated detection of the risk of falling and the estimation of balance metrics using COP measurements collected with the WBB.

3.3.1 Detecting risk of falling

Over thirty percent of older adults aged 65 and above fall at least once a year [3]. Approximately ten percent of these incidents result in serious injuries, which does not only affect physiological, but also the psychological wellbeing of these patients. In their work, *Mertes et al.* [3] present an approach to prevent fall incidents by evaluating a person's risk of falling using machine learning techniques. They collected data from 39 individuals, including 11 people that have declared to have fallen at least once in the last 12 months. The participants performed two exercises on the WBB, each with eyes opened and eyes closed. The first exercise was to place their feet on a predefined outline on the WBB and stand still, keeping an upright stance, also referred to as rigid stance. For the second exercise, participants were asked to press their knees and feet together into a narrow stance. The narrow stance with eyes closed was too difficult for participants which experienced prior falls, hence this particular exercise was removed from the study. Every participant performed three exercises for 40 seconds each, during which COP data was collected. From the gathered data, the mean velocity, RMS velocity and RMS amplitude was computed and utilized as features for a set of machine learning algorithms. *Mertes et al.* trained a Support Vector Machine (SVM) and a k-Nearest Neighbor (kNN) classifier with two configurations each, resulting in four trained classifiers for each exercise.

Support Vector Machine classifier The SVM classifier is a supervised machine learning algorithm, thus learning to classify new data points from a labelled training set. The SVM tries to find the maximum empty space (area without any data points) that seperates classes defined in the training set. In a 2-dimensional feature space the hyperplane, which constitutes the border between classes, is linear. Data points that define the hyperplane are referred to as the support vectors. Figure 3.9 shows two possible linear hyperplanes, A and B, to seperate two classes. In this example, A is the better choice, because it creates a larger margin between the classes. SVMs are also referred to as "*large margin classifiers*".

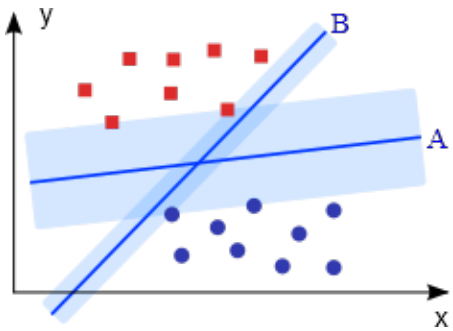


Figure 3.9: Possible linear hyperplanes (A and B) for a binary classification problem [67].

k-Nearest Neighbor classifier The kNN classifier is also a supervised machine learning algorithm. It is parameterized with the number of neighbors (k) to include

during inference. Given a new data point, a distance metric, e.g. Euclidean distance, is used to calculate the distance from the new data point to each point in the training set. Assuming a k-value of 3, the algorithm checks the label (=class assignment) of the 3 neighbors in the training set which are closest to the new data point. If two of the three neighboring data points are of class *C1* and the other neighbor is of class *C2*, the new data point will be classified as *C1*, because the majority of its k nearest neighbors are of class *C1*. Figure 3.10 illustrates how the choice of the k-parameter can alter classification results for kNN classification. With k=3 the majority of neighbors belong to the class illustrated as red triangles. Increasing k to 5 swings the classification results in favor of the class depicted as blue squares.

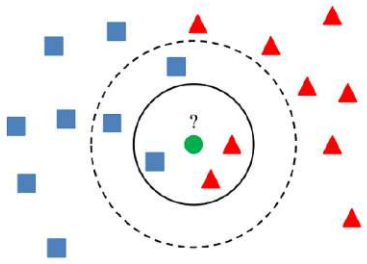


Figure 3.10: Influence of parameter k on classification result for kNN classifier [68].

Figure 3.11 shows the separability of training data using the SVM classifier. Each point represents a participants RMS velocity in medio-lateral and anterior-posterior direction. A clear separation between the classes within the training data can be observed, represented by the linear hyperplane.

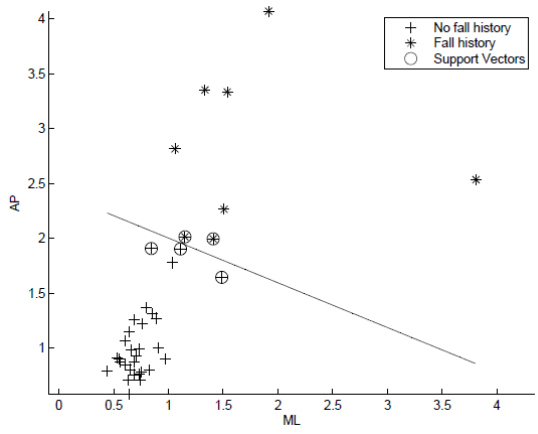


Figure 3.11: SVM training data with support vectors and linear hyperplane for RMS velocity. Exercise: rigid stance with eyes open [3].

Mertes et al. show that RMS velocity and mean velocity perform equally well for the task

of differentiating fallers from non-fallers. Furthermore, they show that RMS amplitude performs the worst for this task. They conclude that WBB sensory data can be utilized to distinguish between fallers and non-fallers. This provides evidence, that data gathered from a WBB can be used to gain valuable insight into the balance ability of patients. Especially the choice of parameters used by *Mertes et al.* is evidently very indicative of balance ability.
The prototype developed in the course of this thesis is utilizing the same COP-based balance parameters that are used in the work of *Mertes et al.*, because of the convincing results established in their work and the research (*Piirtola et al.* [69] and *Melzer et al.* [70]) they base their choice of parameters on.

3.3.2 Estimation of balance ability

Bacciu et al. [9] propose a system that can accurately estimate a person's BBS by utilizing sensory data from the WBB. Estimation of balance ability is usually done by performing exercises of an exercise protocol, e.g. the BBS, followed by grading correctness of execution by a therapist. This process takes, on average, between 15 and 20 minutes. The system proposed by *Bacciu et al.* only requires one out of 14 exercises of the BBS, performed for approximately 10 seconds, to accurately estimate a person's balance ability. Thus, their approach is extremely time-saving and is also described as "*unobstrusive, safe and easy to use even without supervision by clinicians*" [9].
To achieve this, *Bacciu et al.* gathered data from 21 volunteers, aged between 65 and 80 years. "*Turning to look around*" was selected as BBS exercise of choice, because it does not include interaction with any objects, it can be performed on a WBB and *Bacciu et al.* found that it contained richer temporal data compared to standing exercises because of the dynamic nature of the exercise. The gathered sensory data was pre-processed by removing measurements below a weight threshold of 4kg to remove recorded data before a patient stepped on the WBB. Furthermore, filtering techniques were applied to the data, see figure 3.12, to smoothen data and remove unwanted noise.

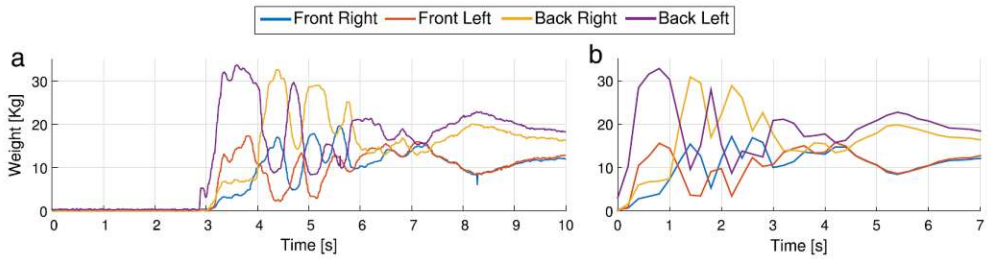


Figure 3.12: Pre-processing of WBB sensory data: a) Before pre-processing, b) After pre-processing [9].

The preprocessed data, assigned with the corresponding BBS, was passed into a Recurrent Neural Network (RNN). A RNN is a supervised machine learning algorithm that is

particularly efficient at finding patterns in noisy temporal data. The schematic flow of the system proposed by *Bacciu et al.* is illustrated in figure 3.13.

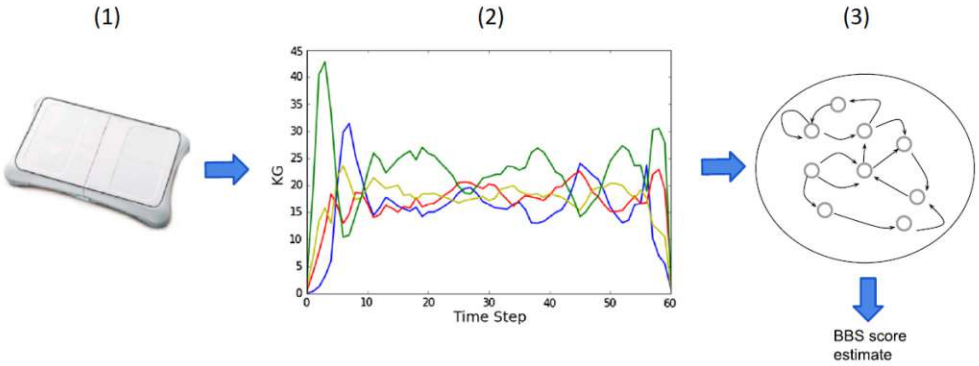


Figure 3.13: The flow of the proposed system for automatic balance score estimation: (1) Data collection using the WBB, (2) Pre-processing of sensory data, (3) Estimation of BBS using RNN [9].

The resulting system is capable of estimating the overall BBS with an absolute deviation from the ground-truth value of 3.80 points, only utilizing the data of a 10 second measurement comprising of only a single BBS exercise. In their work, *Bacciu et al.* do not rely on extracting (static) COP-based metrics from their training data, but rather use a RNN to find the distinguishing features of trials evaluated with varying BBS. Deep-learning techniques, such as the RNN, depend on having a large training set, which is hard to aquire, especially in the healthcare sector. This thesis attempts to model healthy balance ability to subsequently calculate a score for patients' balance ability based on the deviation from this healthy baseline. The purpose of an RNN is to find patterns in streams of data and distinguish between two or more classes. Even though the results provided by *Bacciu et al.* are promising, the proposed approach is not applicable for the problem this thesis is trying to solve. These results show nonetheless that WBB sensory data encodes rich information about a person's balance ability.

3.4 Adaption to player performance

Borghese et al. [8] developed the *Intelligent Game Engine for Rehabilitation (IGER)*, which is a system to support patients' at-home rehabilitation. The relevant components of the system are the hospital station and the patient station. The hospital station is used by clinicians to monitor patient performance, support the patient by interacting with them as a virtual therapist and to schedule therapy sessions for the patients. The patient station, installed at the patient's home, guides them through their rehabilitation session, providing engaging exergames while monitoring exercise execution.
The game engine, illustrated in the system overview in figure 3.14, is at the core of the patient station.

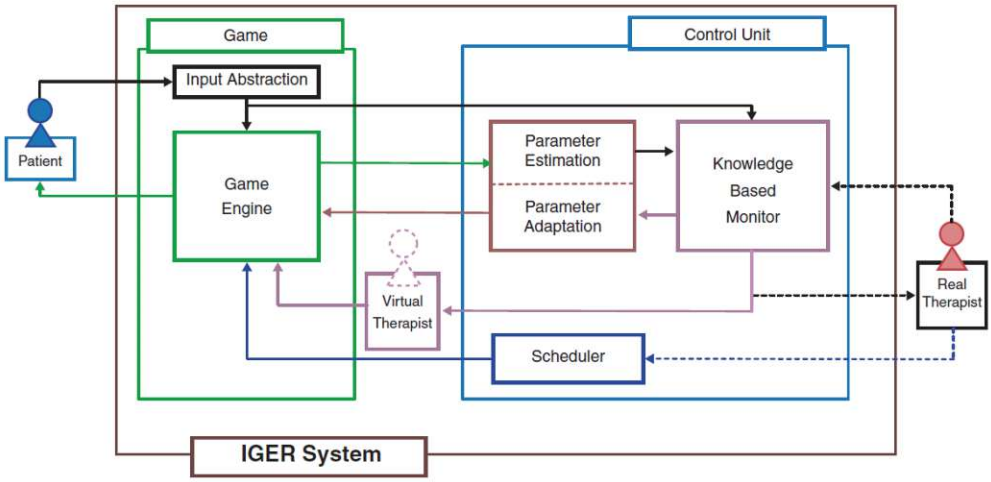


Figure 3.14: A system overview of the Intelligent Game Engine for Rehabilitation (IGER) [8].

It schedules the games, adapts game difficulty based on patient performance, supervises the game sessions and displays the virtual therapist. Adaption of game difficulty based on patient performance is done through parameterization. "*Games defined for rehabilitation have to be parametric. Specifically, games should be defined by parameters that can be regulated and adapted, depending on the level of game difficulty.*" *[8]* These parameters can either be manipulated by the clinician through the hospital station or by the game engine based on performance after a patient completes a game level. Parameter adaption via the game engine does not take sensory data into consideration. It attempts to increase or decrease parameter values in such a way that a certain success rate is maintained.
Furthermore, IGER includes a rule-based monitoring system, where the clinician defines a set of rules, e.g. keeping the feet slightly apart. These rules also include a maximum range allowed for correct execution, depending on the patient's physical constraints. The hospital system raises an alarm if exercises are performed incorrectly.
Figure 3.15 shows two of the games developed for IGER, which highlight IGER's adaptation capabilities. As shown in figure 3.14, IGER has an input abstraction module, which allows it to process input data coming from different source devices, e.g. the WBB or a Kinect. The Kinect uses depth-sensors to enable motion-based controls for video games. "Animal Feeder", illustrated in figure 3.15 (1) contains a balance exercise that requires the patient to kneel in front of a display and feed virtual cows with their impaired arm using the Kinect controller. If a cow is not fed fast enough, the player's score decreases. The WBB's purpose in this game is only for data collection.
The second game, shown in figure 3.15 (2), is called "Fruit Catcher". In this game, the patient has to perform lateral weight shifts to collect virtual fruits falling from a tree. Adaptions in difficulty include changing the number of fruit that drop or the height fruits fall from.

Figure 3.15: *IGER* minigames: (1) *Animal Feeder* and (2) *Fruit Catcher* [8]

Borghese et al. identified adaption of game difficulty based on player performance as a crucial part of at-home rehabilitation, or rather therapy success using SGs in general. Adaptation of level difficulty in IGER is solely based on performance during gameplay. The sensory data collected via input devices, such as the WBB, is only used as additional information for clinicians to make deductions about patients' balance ability and change game parameterization if the difficulty exceeds the patient's skill level.

Current state-of-the-art systems for balance rehabilitation, such as *Gomes-Gil et al.*'s *eBaViR* [7] or *Borghese et al.*'s *IGER* [8], propose sophisticated adaptation mechanisms to support patients with varying levels of skill. These adaptation mechanisms change parameters, such as number or speed of obstacles, based on the player's achieved score. This score increases or decreases based on the player achieving abstract objectives within the game level, such as the number of fruit caught (or missed) 3.15 (2) or the number of balloons collected (or missed) 3.7 (3). Due to the fact that positioning of game objectives, e.g. balloons or fruit, is randomized, this can lead to a misrepresentation of the player's balance ability due to the objectives repeatedly spawning at either the perfect or worst possible locations. It can be argued that the impact of objective spawning locations will average out over time, but it can still affect performance in the short term. Furthermore, a score solely dependent on counting achieved in-game objectives does not incorporate correct exercise execution.
State of the art in balance estimation based on COP data, represented in this section by estimating fall risk [3] or balance ability [9], proves that considerable information can be extracted from the WBB's sensory data stream. State-of-the-art systems for balance rehabilitation that aim to adapt game difficulty based on patient skill gather this information, but do not automatically process it and derive further insight from it. This is a shortcoming of the current state of the art.
This thesis proposes evaluation of patient performance during gameplay by analyzing the WBB sensory data, extracting COP-based parameters from the data stream and measuring the deviation of patient balance ability from a baseline established from

healthy subjects. This also circumvents the problem of acquiring large amounts of healthcare-related data from balance impaired patients.
The work in this thesis builds upon research conducted for SGs in balance rehabilitation, learning balance ability from WBB sensory data and adapting game environments based on player performance. The following section discusses the core contributions of this thesis in relation to the current state of the art.

3.5 Contributions of this thesis

Table 3.1 compares core aspects of the current state of the art and outlines the focus of contributions in this thesis. The main goal is to evolve the current state of the art in adapting game difficulty based on patient impairment. The aspects that are involved in this regard are annotated with an asterisk.

	RehaLabyrinth [33]	**SilverBalance [65]**	**eBaViR [7]**	**WeHab [1]**	**IGER [8]**	**Walk in the park**
Gamification elements	✓	✓	✓	-	✓	✓
Calibration	✓	-	✓	-	-	✓
*Balance evaluation**	-	-	-	✓	-	✓
*Performance feedback**	✓	✓	✓	-	✓	✓
*Difficulty adaptation**	-	-	✓	-	✓	✓
Multiple exercises	-	-	-	✓	✓	-

Table 3.1: Comparison of core aspects with current state-of-the-art work

Firstly, an extensive literature research and interviews with experienced physiotherapists aim to compile a sophisticated catalog of requirements, aiming to incorporate the most vital gamification elements from current state-of-the-art work into the prototype. As a result, this work contributes a snapshot of current state-of-the-art gamification elements in SGs for balance rehabilitation, which is in-line with the first research objective of this thesis.
Incorporating calibration to counteract severe one-sided balance impairments is already commonplace in current state-of-the-art work, thus it is also part of *Walk in the park*.
In the current state of the art for adaptation to player performance by quantified balance ability, defined by eBaViR [7] or IGER [8], performance is derived by the amount of in-game objectives achieved during gameplay. These approaches do not make use of valuable data emitted by the WBB during gameplay. Contributions of *Mertes et al.* [3] and *Bacciu et al.* [9] provide evidence of the richness and expressiveness encoded in COP data. *Kennedy et al.* [1] attempt to utilize WBB-based COP data to compute an overall balance score but found no meaningful correlation between physiotherapists balance estimations and the computed scores. This is where *Walk in the park* contributes to the current state of the art by introducing a novel approach of computing balance scores based on WBB data and machine learning techniques. Furthermore, balance evaluation is performed independently for directional weight shifts, resulting in scores for left-, right, up-, and forwards weight shift performance.
Despite most state-of-the-art contributions providing performance feedback, as stated

above, it relates to achieved in-game objectives, thus is not a direct reflection of patient balance ability. It also does not provide insight into weaknesses in balance ability. Only *WeHab* provides additional balance metrics, derived from COP data, which can be used to make deductions about balance behaviour. *Walk in the park* provides balance score feedback after each completed level. Furthermore, after completing a session, insight into individual balance metrics is provided.
Furthermore, *Walk in the park* contributes a novel way of adjusting game difficulty by generating personalized levels. Level generation is based on requirements gathered during interviews with physiotherapists. This ensures that levels are generated in a way to adequately reflect patient impairments and targets the second research objective. Since level generation is parameterized by patient balance scores, it is highly dependent on the novel approach for quantifying balance ability providing meaningful results.
Walk in the park focuses on a directional weight shifting exercise. Providing support for multiple exercises, as many state-of-the-art solutions do, can be considered in future work if the introduced balance evaluation and subsequent personalized level generation is deemed satisfying.

CHAPTER 4

Results

This chapter presents the outcomes of research and implementation efforts, ultimately resulting in the prototype called *Walk in the park*. It is subdivided into six parts: (1) results of the research phase, (2) results of the design phase, (3) overview of the system, (4) implementation details of the frontend, (5) implementation details of the backend and (6) implementation details of personalized level generation.
Firstly, results of the research phase are presented and discussed in detail in section 4.1, targeting the first and second research objective. The research phase is subdivided in a requirements engineering phase and a technology research phase. During the requirements engineering phase, described in section 4.1.1, interviews were conducted with two physiotherapists, specialized in balance rehabilitation, gathering requirements for SGs in balance rehabilitation, as well as requirements for adapting to patient impairment via personalized level generation, resulting in comprehensive requirement catalogs for each subject. This phase was accompanied by an extensive literature review, resulting in another catalog containing vital design criteria for SGs in balance rehabilitation. Additional requirements were defined by the author to make the system usable for multiple organizations, allow user authentication and other general utility purposes. The requirements engineering phase was followed by a technology research phase, evaluating different combinations of technologies to ultimately decide on the best technological fit for prototype implementation in section 4.1.2.
This is followed by a section outlining the results of the design phase, presenting the high-level mockups and associating design decisions with requirements gathered during the research phase. Both physiotherapists were participating in this phase, iteratively refining the proposed prototype design until producing a final version. The resulting mockups, shown in section 4.2, constitute the blueprint for prototype implementation.
Subsequently, the system overview in section 4.3 aims to provide a general understanding of all system components, how the frontend and backend interact with eachother and a brief section about the hardware used to implement *Walk in the park*.

Implementation details about the frontend, described in section 4.4, and backend, see section 4.5, focus on technologies used for the respective subsystem and in which context they were applied.
Finally, the centerpiece of this thesis, personalized level generation, is discussed in detail, combining frontend and backend functionality, laying the foundation to answer the third research objective. During this phase, data from 11 volunteers was collected to model healthy balance ability. Modeling of balance ability and calculation of patient performance, which are crucial components for personalized level generation, are discussed in more detail in section 4.6.

4.1 Research phase

The research phase is divided into a requirements engineering phase and a technology research phase. Results of this process are pivotal for answering the first two research questions stated in 1.2. The requirements are elicited though extensive literature research and interviews with physiotherapists specialized in balance rehabilitation. The subsequent technology research phase provides evaluation results of technologies considered for the proposed prototype and outlines the system architecture.

4.1.1 Requirements engineering

The first step of requirements engineering is to identify the relevant stakeholders. This thesis has a rather small and clear-cut set of stakeholders, mentioned in the following paragraphs.

Advisors Their main interest is the scientific relevance of the proposed approach and appropriate documentation of work done in the context of this thesis.

Physiotherapists Involved in development of the prototype by providing expert knowledge about balance rehabilitation in practice during all stages of prototype development. Their main objective is the exploration of new strategies for balance rehabilitation.

Volunteers for building healthy balance model A total of 11 volunteers with healthy balance ability participated in data acquisition, necessary to build the model for balance score computation.

Patients Patients with balance impairments are involved in the evaluation process of the prototype. Their incentive to participate is the potential motivational boost in rehabilitation and increased therapy success.

The phase within requirement engineering that follows identification of relevant stakeholders is requirements elicitation, where techniques to identify requirements are applied

and requirements are specified. Requirements for the SG developed in this thesis were elicited with a combination of unstructured interviews and literature research. In the context of this thesis, the requirements engineering phase constitutes a vital part of answering the first two research objectives:

1. What are the requirements for a SG in the context of balance rehabilitation?
2. What are the requirements for personalized level generation in balance rehabilitation?

The following sections document the results gathered utilizing the aforementioned elicitation techniques to ultimately create a catalog of requirements that aim to answer the first two research questions.

Requirements for SG in balance rehabilitation

Requirements elicitation for SGs in balance rehabilitation was done using a combination of document analysis and unstructured interviews, thus requirements were gathered based on both qualitative and quantitative methods. Document analysis constitutes the quantitative part by performing extensive literature research. Qualitative data was acquired by conducting unstructured interviews with physiotherapists specialized in balance rehabilitation. The first step of the literature research was to collect relevant state-of-the-art work for SGs in balance rehabilitation based on a predefined set of inclusion criteria. For this purpose, Google's academic search engine *Google Scholar* [71] was used, which is currently with over 389 million records the most comprehensive academic search engine available [72].

Requirements from literature The first part of the research phase was to conduct document analysis by collecting literature containing design decisions or concrete requirements regarding SGs for balance rehabilitation. This could either be standalone SGs, systems that contain SGs for balance rehabilitation or comparative studies that include information about design decisions when developing SGs for balance rehabilitation. The keywords used for the search on Google Scholar were: "*serious games balance rehabilitation balance board*". Although the input device, in this case the balance board, is not necessarily important for making design decisions about gamification in balance rehabilitation, it can be beneficial to focus on games including the WBB do gain more detailed insight into designing rehabilitation games using force plates, or the WBB specifically.
Performing a search on Google Scholar resulted in 76.000 results, which were sorted by relevancy determined by the search engine. The first 100 results were taken into consideration and analyzed. For the research work to be included in this literature research phase, the papers must have included design decisions or requirements for the proposed SGs. When two or more of the selected scientific contributions mentioned the same or highly similar requirement, they were included into the requirements catalog

shown in table 4.1. The code for each requirement is prefixed with *Lit-* as abbreviation for literature, indicating they are a product of literature research.

Code	Design criterion
Lit-1	Obtaining a valid and adaptive system for the balance rehabilitation of the patients [7] [33] [8] [4]
Lit-2	Achieving a system that reinforces the motivation of the patients during the rehabilitative process [7] [33] [73]
Lit-3	Providing the therapists with objective data of the evolution of the patients [7] [33] [73] [74]
Lit-4	Using simple interaction mechanisms so patients with cognitive impairments can focus on the physical exercise [65] [4]
Lit-5	Establish a relation between the patient's skill level and the challenge level [75] [8]

Table 4.1: Design criteria for SG in balance rehabilitation from literature research

Although part of the goal of literature research phase was to find concrete requirements for SGs in balance rehabilitation, only the work of *Baranyi et al.* [33] lists specific requirements. The other research papers that met the inclusion criteria for this phase defined relevant design criteria instead of concrete requirements for SGs in balance rehabilitation. The specified requirements in the work of *Baranyi et al.* were mapped to the appropriate design criteria gathered from the literature. Thus, the results for this phase contain relevant design criteria instead of concrete requirements that are vital for development of a SGs in balance rehabilitation. The findings are discussed in the following paragraphs in more detail.

Lit-1 A SG for balance rehabilitation has to respect the physical constraints of patients with balance impairments. This can be achieved by measuring the maximum weight displacement the patient is capable of and subsequently adjusting movement behavior within the in-game world accordingly, also referred to as calibration.

Lit-2 *Burke et al.* describe challenge and meaningful play as the two main principles in game design for SGs in balance rehabilitation [4]. Meaningful play emerges from associating the player's actions with the system outcome. The player should be able to perceive how their choices or performance affects the state of the game. Also feedback is a relevant factor of meaningful play, which can be given by scoring mechanisms to quantify performance and visual or auditory cues during gameplay to ensure correctness of exercise execution. This requirement is also achieved by incorporating various gamification mechanics discussed in section 2.1.

Lit-3 Providing physiotherapists with objective data about the patient's performance opens up the possibility to improve therapist understanding of the patient's

impairments. Thus, meaningful adaptations to the therapy routine can be derived from this data to more appropriately suit the patient's needs.

Lit-4 Balance rehabilitation largely affects the elderly and often older patients are not sufficiently familiarized with video games [65]. Thus, keeping game mechanics and level design as simple as possible helps not overburdening patients, which could subsequently lead to them being frustrated and disengaged.

Lit-5 The principle of gradually adjusting the level of challenge posed by the game according to the patient's current performance and level of skill is referred to as *flow theory*. A level that is too easy would lead to boredom, whereas a too difficult level would cause frustration. Adjusting difficulty according to the player's skill level will elevate the patient into a flow state, which in rehabilitation games makes the patient focus only on the challenge, which leads to extended sessions and longer periods of treatment [75].

Requirements from interviews The interviews were conducted with two experienced physiotherapists that are specialized in rehabilitation of patients with balance impairments. The participating physiotherapists are introduced in the following.

Raphaela Borg Head of neurological rehabilitation and physiotherapist at "*aks gesundheit GmbH*", which is a rehabilitation facility specialized in neurological rehabilitation.

Hanna Schlosser Physiotherapist at *SMO Dornbirn*, which is a facility that provides ambulant and full-time neurological rehabilitation.

Both physiotherapists agreed to devote one hour of their time to the interview process. In general, the interviews were conducted in an unstructured manner, but nonetheless with the aim of achieving a mutual understanding of the thesis and deriving requirements from insights into their work in balance rehabilitation. Both interviews started with introducing the aim of this thesis to the therapists and describing the general idea of the prototype to get a mutual understanding. Subsequently, both Raphaela Borg and Hanna Schlosser were asked to provide insight into a typical rehabilitation session with a patient that suffers from balance impairments, irrespective of the underlying pathology with the aim of identifying certain workflows and therapy elements of the balance rehabilitation sessions that can be incorporated into the SG. A compilation of requirements, gathered during both interviews, is shown in table 4.2. The code for each requirement is denoted with *Int-x**, where *Int* indicates association with the interview process. Furthermore, each of the listed requirements is postfixed with an asterisk, because table 4.2 constitutes an intermediary requirement catalog, which is only relevant for the subsequent discussion in section 4.1.1, where design criteria and requirements from interviews are conflated to compose the final requirements catalog 4.3.

Code	Requirement	Weight
Int-1*	Game design and gameplay mechanics should be as simple as possible to accommodate patients with cognitive impairments.	5.0
Int-2*	Incorporate dual task exercises for experienced players.	3.5
Int-3*	After each level the patient should have the ability to pause the game.	4.0
Int-4*	After each level the patient should have the ability to replay the previous level.	4.0
Int-5*	To support patients with severe one-sided balance impairments, adaptation of sensitivity to weight displacement of the balance board (calibration) has to be provided.	3.5
Int-6*	Progress in calibration should be visualized.	4.0
Int-7*	Show feedback of directional balance performance and visualize progress.	5.0
Int-8*	Provide possibility to warm-up before every session.	3.0
Int-9*	Provide tutorial that patients can initially experience and learn the game's workflow.	4.5
Int-10*	Include statistics to provide detailed information e.g. about the number of levels completed.	3.5
Int-11*	The game should provide a game mechanic that forces the player to stand in-place for a certain amount of time.	1
Int-12*	Delay between weight shift and movement in-game should be minimal.	5.0

Table 4.2: Requirements for SGs in balance rehabilitation from interviews with therapists

To analyze the importance of each documented requirement, the collection of elicited requirements was sent to the physiotherapists to assign them an importance. A requirement is considered important to prototype design, if omitting it would drastically decrease utility of the prototype in the scope of a balance rehabilitation session for both the patient and physiotherapist. A value of 1 represents low importance and 5 corresponds to a vital aspect of the prototype. The weight column in table 4.2 constitutes the average importance rating derived of valuations given by both physiotherapists.
Some requirements elicited during the interviews were highly similar to the design criteria found during literature research. In those cases, the requirement from the interview was dropped in favor of the design criteria. Details about consolidation of requirements are discussed in the following section.

Discussion The requirement **Int-1*** was dropped in favor of design criteria **Lit-4**, because both state to keep game design and mechanics simple to compensate for cognitive and visual impairments of elderly patients. Another requirement gathered from the interviews, **Int-2***, was removed because dual task exercises are not compatible with the proposed prototype. Performing additional gestures or cognitive exercises would lead to compromised data, because the introduced distraction or additional movement will influence the COP measurements, thus distort the data. Requirements **Int-8*** and **Int-9*** were combined, because interaction with the game should not require completing

a tutorial first. The warmup phase suggested in **Int-8*** before each session was deemed sufficient. **Int-11*** was dropped from the final requirements catalog because it was considered irrelevant by the physiotherapists. The final requirement catalog for SGs in balance rehabilitation is shown in table 4.3.

Code	Requirement	Type	Source
Int-1	After each level the patient should have the ability to pause the game.	Functional	Int-3*
Int-2	After each level the patient should have the ability to replay the previous level.	Functional	Int-4*
Int-3	To support patients with severe one-sided balance impairments, adaptation of sensitivity to weight displacement of the balance board (calibration) has to be provided.	Functional	Lit-1, Int-5*
Int-4	Progress in calibration should be visualized.	Functional	Lit-2, Int-6*
Int-5	Show feedback of directional balance performance and visualize progress.	Functional	Lit-2, Int-7*
Int-6	Provide possibility to warm-up before every session.	Functional	Lit-2, Int-8*
Int-7	Include statistics to provide detailed information e.g. about the number of levels completed.	Functional	Lit-2, Int-10*
Int-8	Delay between weight shift and movement in-game should be minimal.	Non-functional	Int-12*

Table 4.3: Final requirements catalog for SGs in balance rehabilitation

In addition to these requirements, the design criteria found in the literature research phase are also considered for the implementation of the prototype.

Requirements for SG providing personalized level generation

Requirements regarding personalized level generation were elicited by conducting semi-structured interviews with the physiotherapists. First, a shared understanding of the proposed prototype was created by stating the following ideas:

1. The patient can move through the game level by displacing their weight on the WBB, thus the exercise performed in the prototype is lateral and anterior-posterior weight shifts.
2. The game will be 2D and show the level in a top-down view.
3. A level consists of NxN tiles, where some of these tiles form a path.
4. The goal is to walk to the end of the path by performing weight shifts.

5. Based on patient performance the path will adapt.

Based on this general concept, possibilities for adapting the levels in a meaningful way were discussed during the interviews. Table 4.4 shows the combined outcome of discussions with both physiotherapists, thus the final set of requirements for personalized level generation in SGs for balance rehabilitation. Although these requirements were also gathered during the interview process, to avoid confusion with the requirements for SGs in balance rehabilitation, the code for requirements for personalized level generation are prefixed with *LevGen-* as an abbreviation for level generation.

Code	Requirement
LevGen-1	Levels should be generated in a way that forces patients to shift their weight to the weakest direction (to counteract compensation)
LevGen-2	Generation of levels should be based on historic and current performance data.
LevGen-3	Difficulty of levels should increase steadily during a session.
LevGen-4	Difficulty of levels should slowly lead the patient to their level of proficiency.

Table 4.4: Requirements for personalized level generation in SGs from interviews with physiotherapists

LevGen-1 In balance rehabilitation, patients often try to compensate weaknesses in their impaired limb by putting more weight on their unaffected stronger side. This negatively affects rehabilitation success, because instead of regaining strength in their impaired limb, they become overly reliant on their less-affected limb. Experiences physiotherapists can spot these compensation attempts and counteract by giving feedback or manual correction. In the context of the prototype, this means that levels should be generated to force patients to shift their weight to the weakest direction.

LevGen-2 In order to have continuous progression over multiple rehabilitation sessions and to track rehabilitation success, level generation should take previous patient performance into account. Furthermore, level generation should also integrate progress made in the current session. This allows to always adequately generate levels based on the patient's current state of balance ability. This aligns the prototype with conventional balance rehabilitation where therapy techniques are adapted based on daily form and historic performance of the patient.

LevGen-3 To align with conventional therapy sessions, every session should start at the patient's skill level or below and increase in difficulty as the session goes on. This enables the patient to warm up and reiterate on the mechanics of the game.

LevGen-4 Level difficulty should progress until the patient reaches their best performance. If the score does not steadily increase, the levels will not continue to become more difficult. In conventional physical therapy the therapist knows exactly where the limits for the patient are and they try to get as close to that threshold as possible without overburdening the patient.

Additional requirements by the author

To round out the requirements gathered in the previous stages, additional requirements have been defined to introduce the capability of multiple users, roles and organizations to the prototype. Table 4.5 shows the complete list of additional requirements, which is followed by a brief description of each item. Each requirement is assigned with a code prefixed with *Author-* to indicate the requirement was defined by the author.

Code	Requirement
Author-1	Associate each user with an organization.
Author-2	Add a simple login to authenticate the user against the system.
Author-3	Therapists should be able to add new patients to the system.
Author-4	Provide the possibility to log out of the client application.
Author-5	Provide a patient selection screen before starting a new game session to associate the selected user with the session.

Table 4.5: Additional requirements for the proposed prototype

Author-1 In the context of the prototype developed in this thesis, an organization is the rehabilitation facility the therapist is associated with. This requirement restricts the therapist to the patients that are assigned to their organization, which adds a layer of privacy to the application since a therapist is not able to access any information about patients not associated with their organization. For the SG aspect of the prototype, this also introduces a competitive feature when displaying the patient's overall score alongside their associated organization in the highscore list.

Author-2 To authenticate users against the system, a simple login functionality should be introduced that utilizes login credentials to resolve the user's details, such as role and organizational affiliation. This provides the application with necessary data to unlock certain features associated with the *therapist* role, such as creating a new patient (**Author-3**), and provide role-dependant workflows. Furthermore, authentication creates the possibility of at-home rehabilitation. Since the WBB is an affordable device, patients are enabled to set up the client application and engage in balance therapy sessions from the comfort of their own home.

Author-3 To associate new patients with therapy sessions conducted with the prototype, therapists have to be able to create new patients. In the daily business

of balance therapy institutions, there is a constant stream of new patients, thus this constitutes a vital functionality of the prototype.

Author-4 In clinical practice, the prototype application might be set up on a desktop computer in a therapy room. The prototype is potentially used by multiple physiotherapists on this workstation. Once the prototype application is started and a connection with the WBB is established, this connection is maintained until the client application is closed. By providing a log out functionality, therapists can invalidate their session, yet maintain the WBB connection. This allows for a more fluent transition between sessions.

Author-5 In order to collect data on a patient's therapy success, the patient has to be known to the system (**Author-2**, **Author-3**) and has to have therapy sessions associated with them. This is not necessary if the patient logs into the system by themselves, but if a physiotherapist starts a session with a patient, the patient has to be assigned to the actual therapy session in order to associate them with their in-game performance.

4.1.2 Technology research

During the technology research phase different platforms and software stacks for the SG were taken into consideration and evaluated based on technological fit. This part of the technology research is discussed in the frontend section. The backend section explains technical decisions regarding the utilized backend and machine learning frameworks as well as the database solution.

Frontend

The frontend is the part of a software system that users can interact with. It should be designed with the user's needs in mind. Also technology has to be chosen in a way that maximizes availability across platforms and devices. Technologies that were taken into consideration for the frontend (SG) part of the prototype are web-based (Angular), mobile (Android) and a desktop application (Java). The choice for the specific framework or programming language for each category was based on popularity and personal proficiency. In the following, these technologies are briefly described.

Angular Angular is a powerful web development framework for creating single page applications. Developing the prototype as a web application with Angular offers the ability to support a wide variety of devices, from desktop computers to tablets and even smartphones. The only requirement for any device to run the application is a web browser and internet access.

Android Android is a mobile operating system (OS) by Google based on a modified version of Linux primarily for smartphones and tablets. In 2020 it was the most widely-used mobile OS and thus a suitable candidate for the proposed SG.

Java Java is one of the most popular object-oriented programming languages. Java is platform-independent, which means that every platform, regardless of computer architecture, that supports Java can run Java-based applications. This property makes it possible to compile an application once and run it on different OS without the need or recompilation.

Evaluation Each of these technologies has their advantages and drawbacks for the developing the proposed prototype. Table 4.6 shows a comparison between the mentioned technologies based on technical criteria that are vital for the development process. *Platform dependency* is a measure for how restricted app installation is across different operating systems. *Availability* refers to how simple installation of the app for each technology is. *Deployment effort* refers to the degree of difficulty of releasing or publishing a new version of the application. Finally, *display size* is a technical criteria that is especially important for realizing the game concept.

	Mobile app	**Web app**	**Desktop app**
Platform dependency	high	low	medium
Availability	high	high	medium
Deployment effort	low	medium	medium
Display size	low	high	high

Table 4.6: Comparison of frontend technologies based on criteria vital for prototype development

Platform dependency Android apps are highly platform dependent, since Android apps can only be installed on tablets or smartphones that run on the Android OS. The only viable option to run Android apps on a different system if by utilizing an emulated device. Emulators are virtual representations of systems that replicate system behavior. Installing and configuring Android emulators should not be a precondition for using software, thus the high reliance of an Android app to its platform is seen as a disadvantage. Web apps are platform independent, because they run within a web browser, not on the OS itself. A Java-based desktop application can be installed on different operating systems, such as Linux, Mac or Windows, but not on an Android smartphone because Android uses a customized virtual machine, the Android Runtime (ART), for app execution.

Availability Android apps can easily be distributed via the Google Play store and can be reliably made available for installation worldwide. Since a web app does not require installation, availability is considered high. Desktop apps are usually not hosted by an app repository, such as the Google Play Store, or always readily available by

utilizing a browser. Furthermore, as a prerequisite for running any Java-based desktop app, the system must have the Java runtime installed, thus availability is considered medium.

Deployment effort The deployment process for Android apps is very streamlined. Once configured correctly, new versions of the app can be uploaded to the Google Play store and made publicly available. The process of deploying a web app is less streamlined, since is has to be hosted on a webserver. This webserver has to be monitored and maintained guarantee consistent availability of the app. Deployment of the prototype as desktop app constitutes installing it on a system. To install the app, it has to be downloaded, thus it's archive or executable has to be hosted on a fileserver.

Display size As a SG for balance rehabilitation, the majority of potential players will be elderly patients. To accommodate for vision impairments, screen size should be as large as possible. Android devices are rather small compared to computer screens, thus patients with vision impairments might have problems interacting with the prototype on a small Android device.

Feasibility analysis At the very core of the final decision regarding technical feasibility was interoperability with the WBB. The WBB uses Bluetooth to exchange data with the connected client. The underlying communication protocol is called **L2CAP** (Logical Link Control and Adaptation Protocol).

Android At the start of development (July 2019), Android did not support L2CAP socket connections because of security concerns, thus establishing a connection and exchanging data with the WBB from an Android device was impossible. Consequentially, Android could not be considered for developing the frontend part of the prototype.

Web-based solution A web-based solution would offer the advantages of high accessibility, because whether patients nor therapists would have to download additional software to interact with the system. They would only need a web browser that typically comes pre-installed as part of the OS. As stated in chapter 2.6, to establish a connection or exchange data with a WBB, additional software is needed. The developer community surrounding the Nintendo Wii gaming system has gathered an extensive list of APIs that accomplish this task. Unfortunately none of these libraries are written in Java Script, thus they cannot be bundled into a potential client web application.

Desktop application Finally, communication with the WBB using a Java application was investigated. The default Linux Bluetooth stack *BlueZ* supports L2CAP connections, thus connecting to the WBB and exchanging data can be achieved. Since the desktop application does not require re-implementing the API for WBB-communication, the SG will be implemented as a Java desktop application.

Backend

Although the backend part of a software system is not visible to the user, poor design decisions potentially affect user experience due to increased loading times or corrupted data. The backend's responsibilities of the proposed prototype are: (1) providing information about patients, leaderboards, statistics, etc. (*Data exchange*), (2) providing persistent storage for user and session data (*Data persistence*) and (3) calculating in-game performance based on session data (*Score calculation*). The backend is implemented as a RESTful web service that exposes an API for *data exchange* with the client. *Spring Boot*, a Java framework for bootstrapping server applications, is used to achieve *data exchange*. To store patient data and track their progress over time, their meta data and session data has to be stored. For the purpose of *data persistence*, the document-based database *MongoDB* is used. *Score calculation* is performed by utilizing frameworks and libraries that implement a variety of machine learning models written in the *Python* programming language.

Spring Boot Spring Boot is a framework to bootstrap Java applications. It is built on top of the Spring framework that introduces guidelines and best practices for application design and provides out-of-the-box solutions for common software engineering paradigms, such as dependency injection. Spring applications can seamlessly integrate further Spring modules, such as Spring Data, which allows for an effortless integration of data sources, such as the MongoDB.

MongoDB MongoDB is a general-purpose, document-based (NoSQL) database built for modern applications [76]. It stores data in JSON-like documents. The prototype developed in this thesis has no need for complex querying logic, thus a document-based solution was deemed to be the more natural choice.

Python The proposed system uses machine learning techniques to model balance ability of healthy individuals. Python is the most popular and widely-used programming language in academia and in the industry for tackling machine learning problems [77]. There is an extensive landscape of libraries to choose from, such as *TensorFlow* or *PyTorch*, that provide access to state-of-the-art implementations of machine learning techniques.

4.2 Design phase

Design and implementation of the proposed system were done in parallel to combine the benefits of rapid prototyping with high level mockups. During the design phase, requirements and design principles for SGs in balance rehabilitation were integrated into a game concept and molded into interactive high-level mockups. Firstly, this section explains the general game concept of the SG, followed by a detailed discussion about each screen modeled via mockups. Both of these sections aim to elaborate on how requirements

are incorporated into the final screen designs. Furthermore, to compose a holistic picture of the proposed prototype, the resulting screen flow is presented and discussed. Finally, the last section outlines the system architecture of the proposed prototype.

4.2.1 Game concept

The concept of the game is heavily inspired by SGs discussed in chapter 3, especially by the work of *Baranyi et al.* [33], where players navigate through labyrinth-like game levels by performing weight shift exercises on a WBB. Figure 4.1 schematically illustrates the game concept of the proposed prototype. The tile-based map consists of path- and grass tiles. When moving along the path, the player is constrained to the path tiles. The player, represented by a red dot in figure 4.1, has to move along the path and reach a treasure chest by shifting their weight on the WBB in the respective direction. Path tiles have a direction assigned to them (left, right, up or down). For example, while on a tile assigned to the *left* direction, the player is expected to perform a weight shift towards their left side. The treasure chest at the end of each level holds either a single gold coin or a random weapon. The weapon with the highest value will be automatically equipped at the start of the next level. When armed with a weapon, there is a chance of monsters spawning in the upcoming levels. Also, gold coins can randomly appear on path tiles. These game elements introduce randomness, which can help make the game more enjoyable, as discussed in section 2.1. Furthermore, these random in-game mechanics are a way of measuring the level of engagement with the SG, thus are used to compute a performance-independent highscore metric for the player, which decouples quantified balance performance from the player's rank in the highscore list. This aims to to rate motivated players higher than players with better balance ability to compensate for the dramatic variance in balance ability that patients have and reward perseverance over raw skill. While moving along the path, COP data is collected for each tile. When completing a level, the gathered COP data for each path tile is utilized to evaluate balance performance during gameplay. Collecting data for each direction respectively allows for detailed feedback about weight shift performance. There is two levels of adaptation to tailor the level towards the player's level of skill. Player experience, represented by an ELO rating, defines the number of tiles used to render the level (map size). The second adaptation mechanism is the number of tiles for each direction that is used for path generation, also referred to as path configuration. This mechanism forces patients to predominantly practice weight shifts towards their weaker side, which aims to fulfill requirement **LevGen-1** for personalized level generation, preventing over-compensation by patients, and design criterion **Lit-5**, establishing a connection between the patient's skill and challenge level. Although levels should be generated to force patients to activate their weaker side, to prevent frustration and achieve a system that reinforces patient motivation and engagement, thus align with design criterion **Lit-2**, level generation should also incorporate other directions. The game design represents an adventurous stroll through a park, which gives the prototype its name *Walk in the park*.

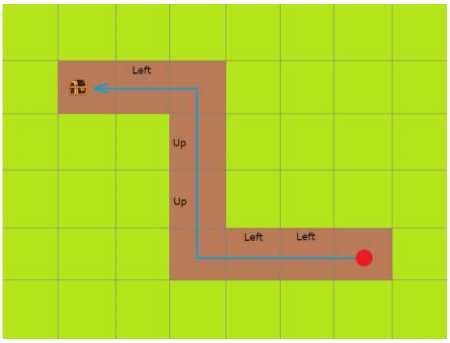


Figure 4.1: Schematic illustration of game concept

4.2.2 Mockups

Mockups constitute the basis for communication and iterative refinement of the prototype. They were hosted online, using the freely available version of *Adobe Experience Design* [78], which allows for creating interactive high-level mockups. To provide clarity and prevent any misunderstandings during the collaborative design process, textual information within the mockups was provided. Sharing these mockups with both physiotherapists resulted in valuable feedback that lead to the final concept illustrated in the following sections. Furthermore, it is described how the design of every screen reflects the design criteria and requirements for SG in balance rehabilitation collected in section 4.1.1. In the following, to create a general understanding, the overall screen flow is presented, followed by an in-depth walkthrough of the individual screens modeled as high-level mockups.

Screen flow

The flow diagram in figure 4.2 describes the screen flow of *Walk in the park*. All screens mentioned in this section are described in more detail in the subsequent sections. After starting the application, the user is presented with the *WBB connection screen*. The user will remain on this screen until the application successfully connects to the WBB, which leads to a redirection to the *Login screen*. Providing invalid credentials will result in error feedback and no screen transition. Entering valid credentials will navigate the user to the *Main menu*. The *Main menu* constitutes the main point of navigation, providing the option to display the *Highscore screen*, create a new patient and start a new *Game* session. When initializing a new *Game* session as a therapist, a patient has to be selected. Starting a new game session as a patient skips this screen. The subsequent screen is the *Calibration screen*. Following the *Calibration*, the user enters the game flow with the *Warmup screen*. The *Warmup screen* presents the opportunity to familiarize with the game mechanics and adapt game parameters, utilizing the *Game settings*. Subsequently, the player can enter the actual game. After completing a level, a *Level summary* provides feedback about player performance. From this point, the player can either decide to repeat the current level, advance to the next level or end the game session. Choosing to repeat the level, the player will stay in the *Game screen* facing the same level again.

Continuing to the next level will generate a new level, also remaining in the *Game screen*. Ending the session will transition the player out of the *Game screen* to the *Session summary*. This concludes the game session by visualizing the player's overall performance and progression. Finally, confirming to have reviewed the *Session summary* will navigate the user back into the *Main menu*.

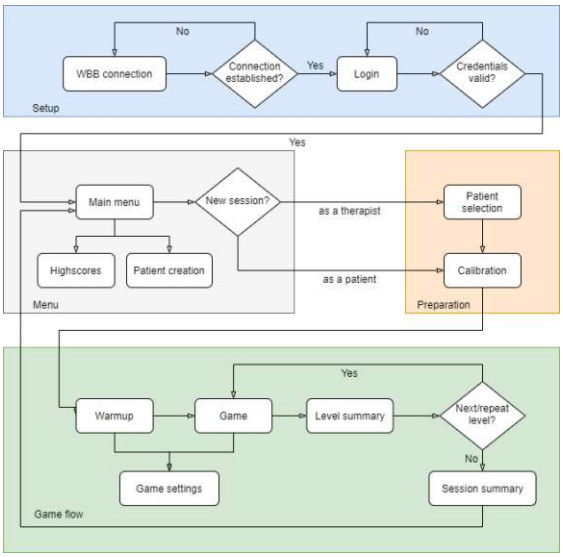


Figure 4.2: Flow diagram of the proposed prototype

Setup

The setup process of the client application consists of two steps: (1) establishing a connection to the WBB and (2) logging into the system.

Connecting to the WBB The WBB has proven to be a viable alternative for laboratory-grade force plates and is utilized as input device in various SGs for balance rehabilitation, see section 3.2. When positioned in close proximity (approx. 2-3 meters), the Bluetooth connection between the WBB and a receiver, e.g. the client application, is shown to be sufficiently low-latency and stable. Thus, utilizing the WBB as input device for the proposed prototype, in combination with an established API to decode the data stream emitted by the WBB (discussed in section 4.4.4), satisfies requirement **Int-8**, guaranteeing minimal delay between the weight shift and in-game movement.
The first interaction with the proposed system is to establish the connection between the WBB and the client, illustrated in figure 4.3. This connection stays open until the WBB is either disconnected or shut down, or until the client application is closed. An advantage of maintaining the connection throughout all sessions is that this step only has to be performed once, thus saves valuable time during rehabilitation sessions.

Figure 4.3: Mockup of the WBB connection screen

Login To authenticate users, a simple login has to be performed, as visualized in figure 4.4. Both therapists and patients have a unique username and password to log into the system, which allows to associate previously stored data with their credentials. This functionality was introduced to comply with requirement **Author-2**, which builds the foundation of associating users with roles, organizations and most importantly, their historic performance data. Providing users with their own credentials furthermore opens up the opportunity for at-home rehabilitation.

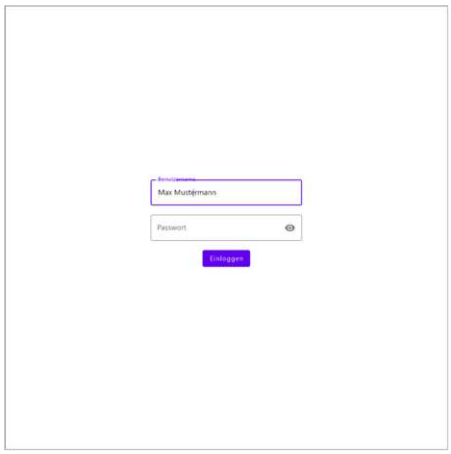


Figure 4.4: Mockup of the login screen

Menu

After successfully connecting to the WBB and logging in, the main screen, shown in figure 4.5, is presented to the user. It displays the logged in user along with four options to proceed: (1) Start a new session (*Session starten*), (2) Show the current high score

list (*Highscores*), (3) Create a new patient (*Neuen Patienten anlegen*) and (4) Log out (*Abmelden*). By selecting (1), the user navigates to the preparations of a new game session. Clicking on (2) navigates to the highscore screen. If the logged in user is a therapist, they are able to see (3), which navigates to the patient creation screen. Lastly, the main screen provides the user the option to log out (4), which navigates back to the login screen. The option to log out was added to satisfy requirement **Author-4**. Invalidating the current user session without losing the WBB connection enables more fluent transitions when switching users.

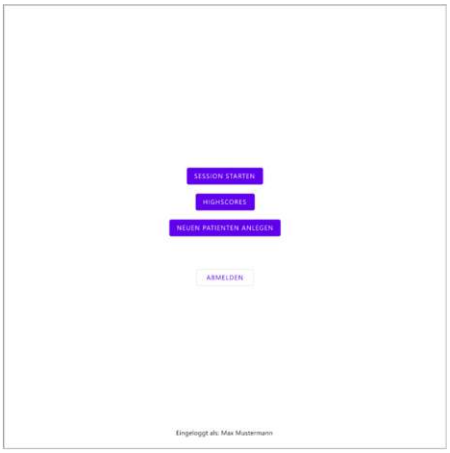


Figure 4.5: Mockup of the menu screen

Highscores To put patient's rehabilitation efforts into perspective, a highscore list is utilized to show patients' usernames along with achievements reached during gameplay. Having means of comparison with other players potentially boosts competitiveness and leads to improved motivation playing the SG. Highscores are also a common gamification mechanic used to display status, discussed in section 2.1. This feature is directly derived from design criterion **Lit-2** for SGs in balance rehabilitation, listed in table 4.1. It also provides therapists with data regarding the patient's rehabilitation efforts, which is captured by design criterion **Lit-3**. Furthermore, the highscore list is based on the concrete requirement for SGs in balance rehabilitation, **Int-7**, because it does not only contain the accumulated score, but also the number of achieved in-game objectives that result in the overall score. The player's rank should be a direct reflection of their rehabilitation efforts, not their quantified balance ability. The reason behind this is varying severity of patients' impairments. Thus, instead of using balance scores achieved during gameplay, the highscore for each player is calculated from the number of collected in-game items that occur at random. The quantified balance ability calculated for each patient is consciously disassociated with the highscores. The highscores should, as mentioned, reflect the time and effort put into rehabilitation, not the actual skill at the game, because of the varying levels of balance impairments that patients suffer from. Using the balance ability directly could lead to frustration in patients that suffer from a

higher degree of mobility constraints.

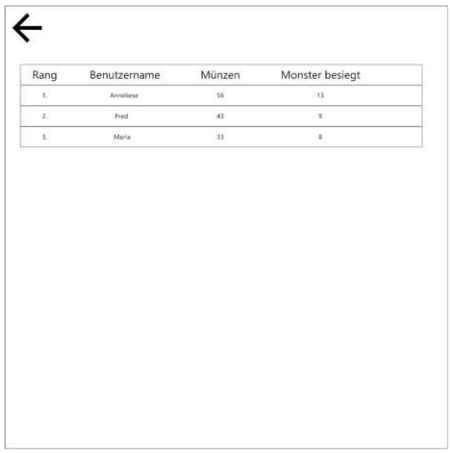


Figure 4.6: Mockup of the highscores screen

Patient creation Creating new patients was implemented to satisfy requirement **Author-3** and can be viewed as basic functionality of the prototype. Thereby, newly created patients are automatically associated with the organization of the responsible therapist, which directly reflects requirement **Author-1**. The benefits of associating users with an organization are described in section 4.1.1. To create a new player, username and password have to be specified, as shown in figure 4.7. These credentials can be used for at-home rehabilitation, since the WBB is reasonably affordable and the proposed prototype does not necessarily require supervision by a therapist.

Figure 4.7: Mockup of the patient creation screen

Preparation

When a new session is started from the main screen, either patient selection of calibration is displayed. If the currently logged in user is a therapist, patient selection is shown. When the logged in user is a patient, this screen is skipped and the calibration screen is shown directly. In general, the preparation phase is parameterizing a new session for a player.

Patient selection To start a new game session for a patient, the patient has to be selected first by entering their username, as shown in figure 4.8. Patient selection, as described in the discussion of requirement **Author-5** in section 4.1.1, is necessary to fetch previous session data of the patient for level adaption as well as initial calibration based on values taken from the previous session.

Figure 4.8: Mockup of the patient selection screen

Calibration After the system knows which patient to associate with the current session, a calibration has to be performed. The purpose of calibration is to make the system adapt to patients with severe one-sided balance impairments, which directly relates to requirement **Int-3**. Post-stroke patients often suffer from hemiparesis, which is a weakness of one entire side of the body. In order to achieve postural stability, these patients need to put more weight on their less impaired limb, thus the center of pressure will, per default, be off-center. To compensate for these severe one-sided impairments, calibration is performed, to correct the center of pressure towards the center. This feature is based on design criterion **Lit-1**, which aims to obtain an adaptive system for balance rehabilitation. Furthermore, requirements **Int-3**, describing the need for calibration for patients with severe one-sided impairments, and **Int-4**, calling for visualizing progress in calibration over time, are directly related to this functionality. The calibration process is illustrated in figure 4.9, where the black outer border depicts the maximum range of COP displacement with an inscribed coordinate system, the red line constitutes the traced COP measurements and the red dot the current COP position in the coordinate system.

To perform calibration, the player has to stand upright on the WBB in a neutral position and attempt to move the dot near the center of the coordinate system. If moving the dot towards the center is achieved using only minimal weight displacement, no calibration has to be performed. If the patient is in a neutral standing position and the dot is far off center and can only be moved using excessive weight displacement, the COP value from the neutral standing position is used as new center of the coordinate system. In-game movement will be calculated based on the calibrated COP data. Furthermore, these calibration values are stored for each patient. This enables tracking progress of calibration over time, which implements requirement **Int-4**. This functionality is displayed in the row above the COP visualization in figure 4.9. By pressing the "OK"-button, the current calibration values are accepted. Positive progress of calibration means that, in a neutral standing position, the distance of the calibration value to the actual center of the coordinate system decreased. After the calibration is done, the game flow can be started by pressing the *Warmup starten* button.

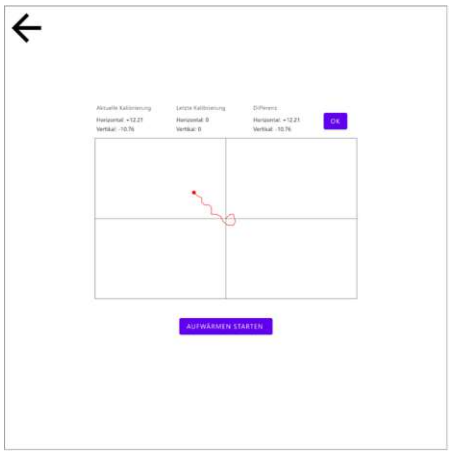


Figure 4.9: Mockup of the calibration screen

Game flow

The game flow mockups already contain the actual game design of *Walk in the park*, because the final iteration of mockups was created in parallel with prototype implementation. The game flow constitutes the core of the prototype, containing the warmup, game settings, the actual gameplay and level design, as well as the level and session performance summaries.

Warmup The first part of the game flow is the warmup. Before starting the game, players can familiarize themselves with the game mechanics by performing weight shift on the WBB and moving through the in-game world. This warmup process before each session is captured by requirement **Int-6**. In this stage of the game, no measurements are recorded. It is meant to learn or re-iteratate on how medio-lateral or anterior-posterior weight shifts are reflected by the position of an in-game avatar, depicted as a character

in figure 4.10. The player can move freely within the bounds of the level. After getting sufficiently comfortable navigating through the warmup/tutorial level, the player can start the game by moving to the area marked with a red arrow.

Figure 4.10: Mockup of the warmup screen

Game settings As shown in figure 4.11 the only properties that can be adjusted in the game settings are directional velocities. If patients can barely shift their weight in a specific direction their avatar will barely move. These settings can be used to speed up or slow down the movement of the in-game character in any direction to prevent loss of motivation during gameplay. Increasing player speed does not influence the COP measurements utilized to calculate the player's balance performance. Enabling tweaking player speed makes the system more adaptable to patient impairments, which corresponds to design criterion **Lit-1**.

Figure 4.11: Mockup of the game settings dialog

Game During actual gameplay, the player has to perform medio-lateral and anterior-posterior weight shifts to move an avatar through a game level. The interaction with the game should be as simple as possible, according to design criterion **Lit-4** to enable patients with cognitive impairments to focus on the exercise. The 2D levels consist of a grass field with a dirt path that the player has to follow in order to reach a treasure chest at the end of the path. An example of a level is depicted in figure 4.12. Various monsters can roam along the path that the player can slay. These monsters spawn randomly and constitute an additional objective within the game, which can boosts engagement by introducing the objective of securing the path (**Lit-2**). Since correct execution of the weigh shifts in each direction is at the core of gameplay, to accurately evaluate directional balance ability, it is sufficient to move onto the same tile as the monster to slay it. The player should not have to perform additional movements towards the monster to slay it, since this could influence the gathered data. The path is generated based on data of previous sessions as well as the current session. The generated path consists of tiles. Each tile is visualized by a square of dirt road. Tiles are directional, which means that for each tile, a specific movement pattern is expected, e.g. a left tile expects a medio-lateral weight shift towards the left side. This enables gathering direction-aware COP measurements along the path and thus distinguish between weight shifts towards the left and right, as well as front and back. At the end or each level, the performance while executing the weight shifts is evaluated and displayed in a level summary.

Figure 4.12: The game screen

Level summary The level summary, illustrated in figure 4.13, shows how well the patient performed in comparison to their previous average results. This data constitutes an objective measure of balance ability, broken down by direction of weight shifts, which provides the physiotherapist with objective data about the patient's balance ability (**Lit-3**). This screen stops gameplay to let the patient and therapist review the level. This period is an inherent pausing mechanism after each level and gives the patient the control when to pause the game, which directly relates to requirement **Int-1**. Furthermore, after

evaluating patient performance, the level can be replayed (**Int-2**) or a new level can be generated that will incorporate patient performance of previous sessions as well as the current session, which establishes a clear relation between the patient's skill level and the level of challenge the patient is facing while playing the game (**Lit-5**). Furthermore, requirement **Int-5** refers to showing feedback of directional balance performance and progress visualization, which is the main purpose of this screen.

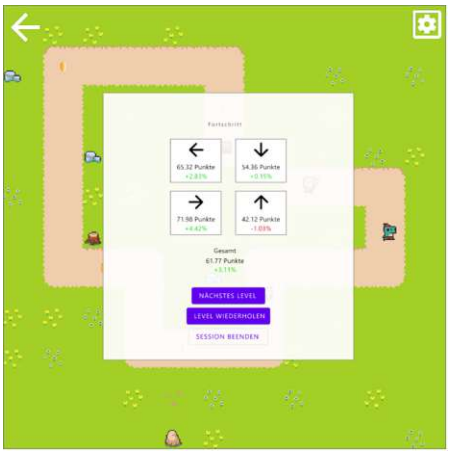


Figure 4.13: Mockup of the level performance screen

Session summary The session summary, illustrated in figure 4.14, shows an aggregation of all levels played during the current session by averaging the results and showing the progress in comparison to the historic average of the player. Just as the level summary, it provides the physiotherapist and the patient with objective data about the patient's balance ability, which reflects design criterion **Lit-3**. The session summary is shown when the session has concluded. In addition to **Lit-3**, this screen also visualizes overall player progress described by requirement **Int-5**.

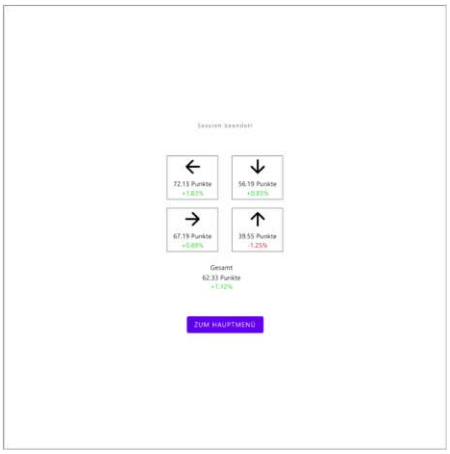


Figure 4.14: Mockup of the game session summary screen

4.2.3 System architecture

Walk in the park is composed of two subsystems, the client (frontend) and the server-side (backend), also referred to as client-server architecture. This architecture is defined by a number of clients connecting to a central server over a network (or internet) connection. The central server manages most of the resources and exposes services that the client can utilize to consume information. The system architecture is illustrated in figure 4.15.

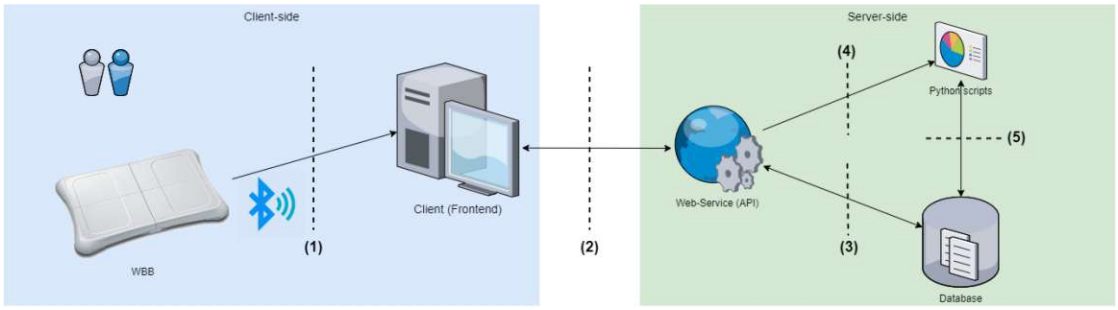


Figure 4.15: Architecture of the proposed prototype

WBB The WBB acts as input device for the SG. During balance rehabilitation sessions, patients stand on the WBB and navigate through game levels by performing weight shifts. The weight detected at the four force transducers of the WBB is broadcasted over a Bluetooth connection. This data is received by the client for further processing. More details about how this connection is established and which technologies are used in the process can be found in section 4.4.4.

Client (Frontend) The client's responsibility is to maintain a stable connection to the WBB and run the SG aspect of the prototype. The game is ideally displayed on a

large display to better support patients with impaired eye-sight.

Web-Service (API) The main responsibility of the API is to provide the client with the necessary data. In addition to accessing the database for information, a suite of Python scrips can be executed that performs score calculation.

Database The MongoDB database is used as persistent storage solution. It contains login credentials for patients and therapists, session data with COP values and serialized versions of the trained machine learning models.

Python scripts A suite of Python scripts is used to quantify balance ability and calculate a performance score for a session, utilizing COP values of a session and trained machine learning models loaded from the database.

Implementation details about the communication interfaces between the system components, depicted in figure 4.15 as **(1)**, **(2)**,**(3)**, **(4)** and **(5)** are discussed in section 4.3.1.

4.3 System overview

This section briefly outlines the responsibilities of each component of *Walk in the park* and describes the implementation details of the client-server communication. The goal is to provide a general understanding of the system and how the components interact before elaborating on implementation details of the frontend and backend in the subsequent sections.

4.3.1 Component responsibilities

The components of *Walk in the park* are the WBB, the **client** application, the **webservice**, the **database** and the **Python suite** for score calculation, shown in figure 4.15.

WBB In the context of the game, the WBB is the input device to move through the in-game level. The sole responsibility of the WBB is to provide the client application with sensory data measured at the four force transducers. This data is translated into COP measurements, stored in-memory and finally transmitted to the backend for persistence and score calculation.

Client (Frontend) The client is the main element of user interaction. Its core responsibilities are (1) estabishing a stable connection to the WBB, (2) authenticating users to access the system, (3) managing session data during gameplay, (4) sending gathered data to the server, (5) receiving data from the server to provide performance feedback and (6) providing a motivating and engaging SG for the patients.

Web-Service (API) The API constitutes a gateway to the data storage and the machine learning-based score calculation. Furthermore, based on the client request, additional data manipulations are applied.

Database The MongoDB database has the sole responsibility of storing data, by either storing updated model data triggered by the Python suite **(5)** or persisting data received from the client application **(3)**.

Python suite The Python suite bundles a variety of functionality to visualize and explore COP data. The part that is relevant for the backend system is model inference, which is the process of utilizing the model data, stored in the MongoDB database, to evaluate new COP measurements and calculate a balance score based on them. Model training is only performed once before the system is deployed to have consistent results across multiple sessions.

In addition to the components involved in *Walk in the park*, figure 4.15 depicts the directed communication channels between them. The implementation details of establishing a connection to the WBB **(1)** are described in section 4.4.4 in more detail. Internal communication between the server-sided components, visualized by the dashed lines **(3)**, **(4)** and **(5)**, is discussed in section 4.5. To establish a general understanding of the overall system, an introduction of the data that is exchanged between the client-side and the server-side **(2)** is provided in the following section.

4.3.2 Client-server communication

The webservice exposes an API for the client to send and receive information from the backend system, which adheres to REST (**Re**presentational **S**tate **T**ransfer) principles, thus is considered a RESTful webservice. For a RESTful webservice any data is considered a *resource* that the client application can fetch, update, create or delete. These resources can be accessed by the client by sending a *request* to a specific URL to which the webservice answers with an appropriate *response*. This can be considered a contract between the client application and the webservice. Below is an example for a URL to access the *Session* resource of *Walk in the park*. It specifies the protocol that is used to access the resource, which is the HTTP protocol, followed by the server address (*localhost*) and port *8080* that is used to access the API. This part is necessary to identify and locate the webservice, whereas the latter part */api/session* is the actual location of the *Session* resource within the webservice.

http://localhost:8080/api/session

REST is stateless, which means that the client has to send all the information needed for the server to process the request. Data exchange via REST webservices is done via the HTTP protocol, which specifies request methods (or HTTP verbs) such as GET (fetch), PUT (update), POST (create) or DELETE. Thus, to properly specify a request to the RESTful webservice, in addition to the URL, locating the resource, the appropriate

HTTP verb has to be specified. The following example specifies fetching *Session* data by prefixing the URL with the GET HTTP verb.

GET http://localhost:8080/api/session

When further information is necessary, query parameters can be sent in addition to the URL to provide more context to the backend. In the following example, we are interested in fetching (GET) *Session* data of the patient with *patientId=123*.

GET http://localhost:8080/api/session?patientId=123

Data is transfered using the JSON format, thus both frontend and backend have to be capable of serializing model data into a valid JSON format and, vice versa, deserializing JSON into model data. Listing 4.1 shows the *Session* model class, whereas listing 4.2 shows a representation of a *Session* object serialized as JSON document.

```
public class Session {
    String id;
    String therapistId;
    String patientId;
    long start;
    long end;
    CalibrationDataDto calibrationData;
    List<String> levelData;
    EvaluationResultDto evaluationResult;
}
```

Listing 4.1: Session model class

```
{
    "id": "123",
    "therapistId": "therapistUserId1",
    "patientId": "patientUserId2",
    "start": 1312983127398,
    "end": 13318283170,
    "calibrationData": {
        copXDiff: 0.0,
        copYDiff: 0.0
    },
    "levelData": {
        "levelDataId1"
    }
}
```

Listing 4.2: Session JSON example

After establishing a general understanding of the interaction between the client application and the webservice, the following section introduces the API contract and the data model developed for *Walk in the park*.

4.3.3 Data model

Figure 4.16 shows an UML diagram containing all DTO classes utilized during server-client communication. *User*, *Session*, *LevelData* and *HighscoreItem* constitute resources exposed by the RESTful webservice and are described in more detail in the following paragraphs. A larger-scale version of the data model is depicted in figure A1, located in the appendix.

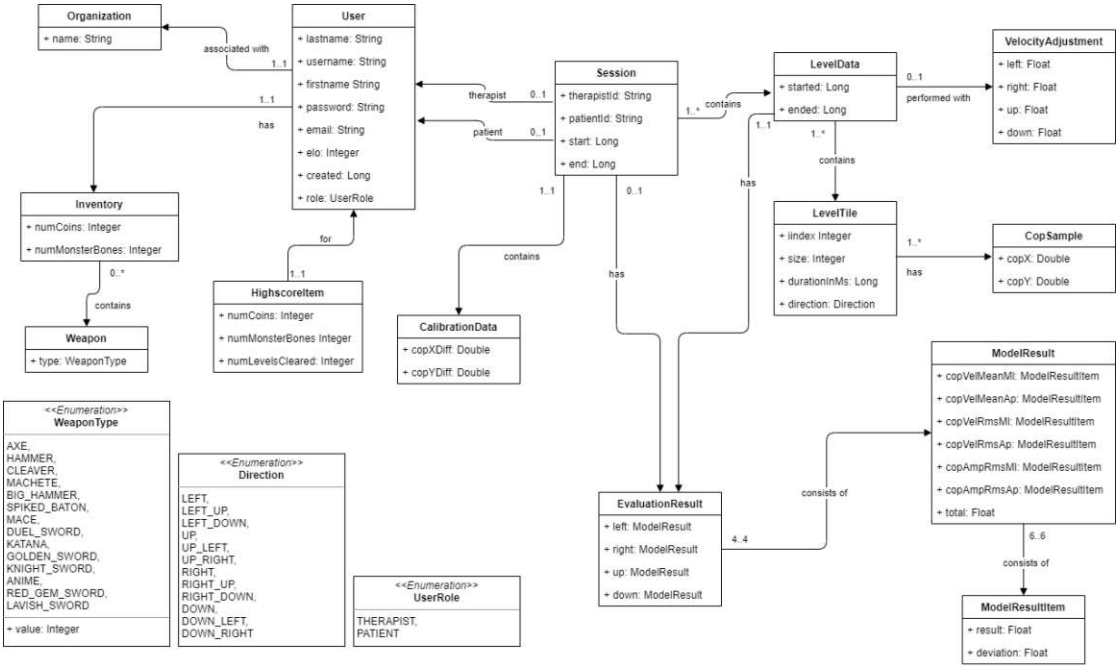


Figure 4.16: Data model of *Walk in the park*

User This class represents people that can interact with the system. *Walk in the park* has two roles, the *therapist* and the *patient*. The therapist has elevated rights within the system, as they can create new patients and start sessions for patients that are assigned to their organization. Assigning therapists and patients to organizations enables the therapist to only access data of their own patients, instead of having access to the whole playerbase. This is a vital aspect for clinical use of *Walk in the park* and prevents leaking patient data. Association of users to an organization is captured in requirement **Author-1**, defined in section 4.1.1. Patients can only start sessions for themselves. Other than their role, the User class contains login credentials (*username* and *password*), the name, email address and creation date. In addition to this basic information, this class also contains the user's experience score (*elo* property), which is a quantification of the their performance playing the game, and their inventory. The inventory contains information about in-game achievements, such as the number of coins collected, monsters slain and weapons found.

Session A *Session* represents a game session that associates the involved users with the levels that were played (*LevelData*) during a session. The *Session* also contains *CalibrationData* and the *EvaluationResult* for the entire session. The *CalibrationData* represents the adjustment applied to user's COP data during gameplay. A calibration is needed in case users suffer from severe one-sided balance impairments that would otherwise prevent them from being able to play the game. This is tied to the session to track progress in calibration data across sessions. Furthermore, the *EvaluationResult*

contains overall balance performance during the session, averaging balance metrics across all completed levels.

LevelData As the name suggests, this class contains the data collected while completing a level. It contains *LevelTiles*, which refer to the path segments within the game that the player navigates through. For each *LevelTile*, a list of COP samples is recorded. This sample data is crucial to the evaluation of balance ability performed for each level. The *direction* property of each *LevelTile* enables evaluation of player's weight displacement for each individual direction, thus investigate weaknesses in balance ability. *EvaluationResults* are calculated per level, not only based on the *Session* to achieve fine-grained intra-session performance feedback. An *EvaluationResult* consists of four *ModelResults*, one for each direction (left, right, up and down). *ModelResults* contains sophisticated COP-based balance parameters, calculated based on the recorded COP samples associated with the *LevelData*. The chosen COP-based metrics for evaluating balance ability are the mean velocity, RMS velocity and RMS amplitude of the COP in both medio-lateral and anterior-posterior direction. Section 2.6.2 discusses each of these COP-based parameters in more detail. In addition to the balance performance result, the deviation of the parameter to a healthy baseline that defines the underlying model is stored in the *ModelResultItem*.

HighscoreItem The *HighscoreItem* contains all necessary information to calculate the player's score based on in-game achievements, their inventory and current experience score.

4.3.4 API contracts

Based on the data model and RESTful design principles for a webservice discussed in the previous sections, an API contract was defined to model server-client communication. Table 4.7 contains all endpoints that the REST webservice exposes for the client application.

HTTP	URL	Description
GET	*/api/users?organizationId=&role=*	Request a list of users from a specified organization and a defined role (THERAPIST or PATIENT).
POST	*/api/user*	Creates a new *User* based on the JSON-encoded data send within the HTTP body.
PUT	*/api/user*	Updates an existing *User*.
GET	*/api/session?patientId=*	Returns the latest *Session* for a specific *User*.
GET	*/api/sessions?patientId=*	Returns a list of *Sessions* for a specific *User*.
POST	*/api/session*	Creates a new *Session* based on provided *Session* data sent via the HTTP body.
POST	*/api/session/sessionId/leveldata*	Adds new *LevelData* to a specified *Session* and responds with *LevelData* containing the *EvaluationResult*.
PUT	*/api/session/sessionId/end*	Finishes open *Session* by computing associated *EvaluationResult* and setting end timestamp.
POST	*/api/login*	Basic authentication with the *User's* username and password.
GET	*/api/leveldata?patientId=*	Retrieve latest *LevelData* for specific *User*. This is utilized by the Python suite to compute the *LevelData's* score.
GET	*/api/highscores*	Returns list of *HighscoreItems*.

Table 4.7: API contracts for *Walk in the park*

4.3.5 Development environment

For the development of *Walk in the park*, a notebook running Linux Mint 19.2 was used. The hardware specifications of the notebook include an Intel Core i7-4719MQ CPU with 8 logical cores, a Nvidia GeForce GTX 860M and 8GB of RAM. Using Linux as operating system has the benefit of utilizing the *BlueZ* Bluetooth stack that is compatible with the software library used to exchange data with the WBB described in section 4.4.4.
The IDE (Integrated Development Environment) used for developing all parts of the system is JetBrains IntelliJ IDEA Community Edition (version 2019.2). JetBrains IntelliJ IDEA is a very potent IDE that offers a variety of features that boost productivity during software development, such as intent-based code-completion, on-the-fly code analysis and reliable refactoring tools [79]. Its plugin-based support for various programming

languages enables implementing all aspects of *Walk in the park* within a single IDE. For both, frontend and backend, OpenJDK version 11.0.3 was used, which provides a free and open-source implementation of the *Java SE Platform Edition*.

4.4 Walk in the park - Frontend

This section describes implementation details about the SG aspect of *Walk in the park*. First, frameworks and libraries are briefly described and how they were used during development. Subsequently, a high-level description of the project structure and architecture is given, followed by a walkthrough section, describing implementation details for each screen. Finally, the approach for personalized level generation, translation of COP measurements to player movement and performance feedback are laid out.

4.4.1 Frameworks and libraries

The main responsibilities for frameworks and libraries used in the frontend aspect of *Walk in the park* are (1) simplifying game development efforts and (2) enabling communication between the client and the WBB. In the following sections, the game development framework and its predefined project structure are discussed. The subsequent section 4.4.4 describes how *WiiRemoteJ* is used to establish a stable connection to the WBB.

LibGDX

Version 1.9.13

LibGDX is a Java-based game development framework with a unified API that works across a variety of platforms. It does not impose any specific design choices during development, thus allows for rapid prototyping and fast iterations [80]. It provides sophisticated features to simplify game development, such as high-level 2D and 3D APIs, audio streaming and tile-map support. LibGDX provides a simple setup wizard that creates *Gradle* powered skeleton project that can be ran, deployed and debugged out-of-the-box.

Gradle

Version 5.4.1

Gradle is a dependency management and build system [81]. External libraries that the projects depends on are specified in a configuration file (*build.gradle*). When triggering the build process, Gradle verifies that these dependencies are available for compilation. If Gradle is not able to locate the necessary dependencies, they are downloaded from a central repository and stored locally. A build system helps with building and packaging an application, without being tied to a specific IDE.

4.4.2 Project structure

The project scaffold for *Walk in the park*, created by the LibGDX setup wizard, contains two modules, the *core* and *desktop* module. The *desktop* module solely contains the launcher class for the desktop application. The code for the game logic resides in the *core* module.

The desktop module

The desktop module only contains the application's launcher class *DesktopLauncher*, shown in listing 4.3, which constitutes the entry point of the game.

```
public class DesktopLauncher {
    public static void main (String[] arg) {
        Lwjgl3ApplicationConfiguration config = new Lwjgl3ApplicationConfiguration();
        config.setWindowedMode(960, 960);
        config.setResizable(false);
        config.setForegroundFPS(100);

        new Lwjgl3Application(new WalkInTheParkGame(), config);
    }
}
```

Listing 4.3: Entry point of the application

This piece of code shows that a configuration of type *Lwjgl3ApplicationConfiguration* as well as a new instance of *WalkInTheParkGame* is passed to the newly instantiated *Lwjgl3Application* object. Lwjgl3Application refers to a **L**ight**w**eight **J**ava **G**ame **L**ibrary (version **3**) application, which LibGDX utilizes to enable multi-platform access to native low-level APIs. LibGDX adds a layer of abstraction on top of game libraries (or backends), such as the LWJGL, to simplify game development. The injected configuration constraints the game window to a fixed (non-resizable) size of 960x960 pixels and a maximum frames per second (FPS) count of 100. Initializing the Lwjgl3Application parameterizes an application using the LWJGL3 backend and starts the game loop with an instance of *WalkInTheParkGame*.

The core module

The core module contains all of the game logic, screens, dialogs, assets, etc. *WalkInTheParkGame* is the entry point to the game logic. Listing 4.4 shows how the game is initialized with a specific screen.

```
public class WalkInTheParkGame extends com.badlogic.gdx.Game {
    @Override
    public void create() {
        setScreen(new LoadingScreen());
    }
}
```

Listing 4.4: Entry point to the game logic

In LibGDX, the *Game* is the top-level entity, which inherits from *ApplicationListener* and delegates to a *Screen* object, in this case an instance of *LoadingScreen*. The create-Method is called only once the first time the application is created. In the case of *Walk in the*

park, an application is started as shown in listing 4.3, with the concrete instance of *WalkInTheParkGame* as ApplicationListener, which then delegates to *LoadingScreen* (WBB connection screen) to finally render the contents of the screen. All of the designs described in the design phase, see section 4.2, are implemented as *Screens*.

4.4.3 Project architecture

This section provides insight into the software architecture of *Walk in the park*'s frontend implementation, utilizing the project's package structure to outline dependencies and associate them with their respective layers. A visualization of the project architecture is depicted in figure 4.17.

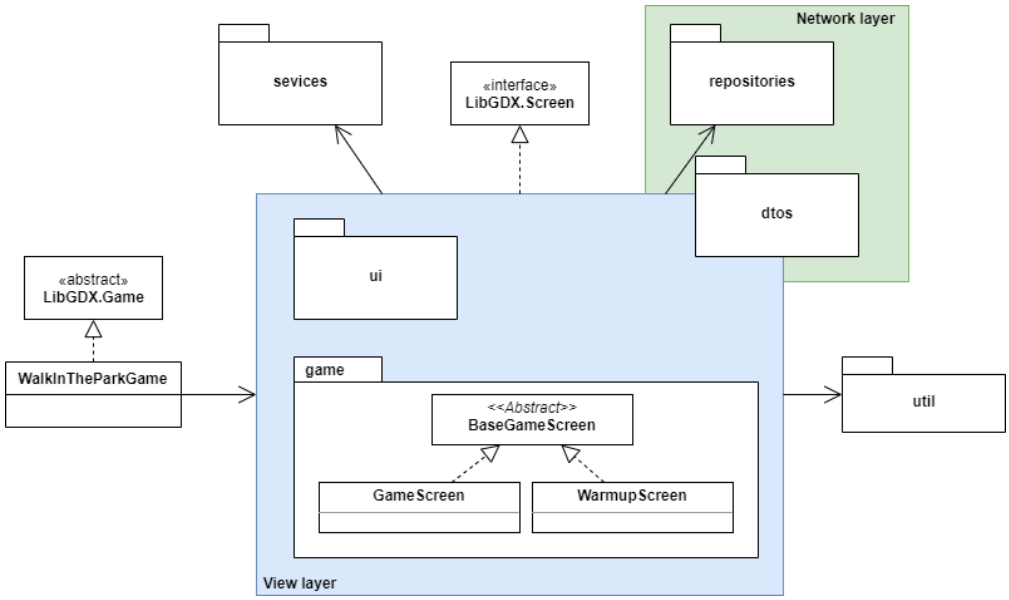


Figure 4.17: Frontend architecture outlined using package structure

WalkInTheParkGame is the entry point into the application, which delegates to *Screens*. The *Screens* actually render the content that is shown to the user, thus constitute the user interface (UI) of the application, also referred to as the **view layer**. Most of the Screens are in the **ui** package.

ui package The classes in this package encompass functionality that deals with WBB connection, authentication, user management and session handling. It constitutes the basis for a system that is suitable for clinical practice and pushes beyond the bounds of a SG. Screens not in the *ui package* are game-related screens, specifically the *GameScreen* and the *WarmupScreen*, because these contain additional game logic.

game package All components of the game logic reside in the *game package*. It contains all classes relevant for the SG aspects of *Walk in the park*.
As illustrated in figure 4.17, both *WarmupScreen* and *GameScreen* inherit from a common *BaseGameScreen* class, which provides shared functionality, such as rendering the level and the player. Since *Walk in the park* is a tile-based game with a top-down view, both *WarmupScreen* and *GameScreen* utilize the *OrthagonalTiledMapRenderer* provided by *LibGDX*. To render a *TiledMap* that represents either the *WarmupScreen* or *GameScreen*, at least one layer has to be added to a *TiledMap* instance. Associating this *TiledMap* with the *OrthagonalTiledMapRenderer* makes it possible to render the contents of *TiledMap*. A high-level overview of how the concrete implementations of *FloorLayers* (*WarmupFloorLayer* and *GameFloorLayer*) interact with *LibGDX* classes is shown in figure 4.18. Both *WarmupFloorLayer* and *GameFloorLayer* hold an instance of *TiledMapTileLayer*, which encapsulates all information necessary to render either the warmup or a game level. Every *TiledMapTileLayer* contains a 2D array of *Cell* objects that wraps the actual *TiledMapTile*. The *Cell* is used to perform additional transformations to the tile's texture, such as flipping or rotation. The *TiledMapTile* represents a single tile within the layer. *StaticTiledMapTile* represents an implementation of *TiledMapTile* that does not change, as opposed to *AnimatedTiledMapTile*. As the textures utilized for *Walk in the park* are static, the *FloorLayerTile* class, which represents a single tile in both the *WarmupFloorLayer* and the *GameFloorLayer*, extends *StaticTiledMapTile*.

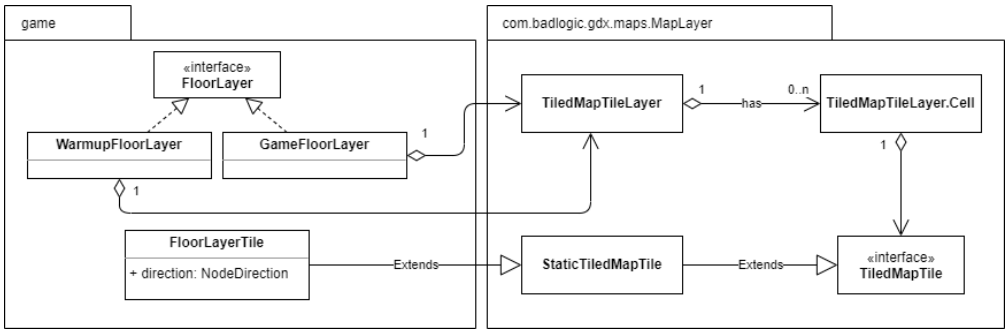


Figure 4.18: Tiled map rendering class interaction

services package The **services package** contains classes for establishing a connection to the WBB and managing the current login session. The former is explained in more detail in section 4.4.4. Login session management is provided by the *ApplicationSessionService* class, which is a singleton service that keeps track of the logged in user and the currently selected patient of a session.

repositories package The **network layer** is used to send and receive data from the *Walk in the park* backend. It contains a set of repository classes in the **repositories package** that implement the API contracts defined by the backend. Thus, the network layer constitutes the communication channel between frontend and backend.

dtos The **dtos package** (**D**ata **T**ransfer **O**bjects) contains the model classes that are used for network communication and within the context of the game. They implement JSON serialization and deserialization of the model classes for network transmission, as well as additional methods based on their role within the game. This dual functionality of the DTO classes places them on the intersection of the view and the network layer.

utils The **util package** contains relevant constants used throughout the application and convenience methods in the *AssetManager* class to load assets used in the game.

The heart of every game is the visual representation and interaction with the UI, thus most of the implementational details are tied to the view layer, which is implemented mostly by using LibGDX Screens. The following sections describe each screen in the **ui** and *game packages* to showcase their implementational details and if applicable, to reflect how requirements were embedded during implementation.

4.4.4 Prototype walkthrough

This section presents the implemented screens of the *Walk in the park* prototype, referencing to each individual screen from the design phase while providing relevant implementation details.

Screen flow

Figure 4.19 shows how the initial design of the screen flow, presented in section 4.2.2, is translated into the prototype implementation of *Walk in the park*. It incorporates all screen implementations discussed in this section and visualizes screen transitions.

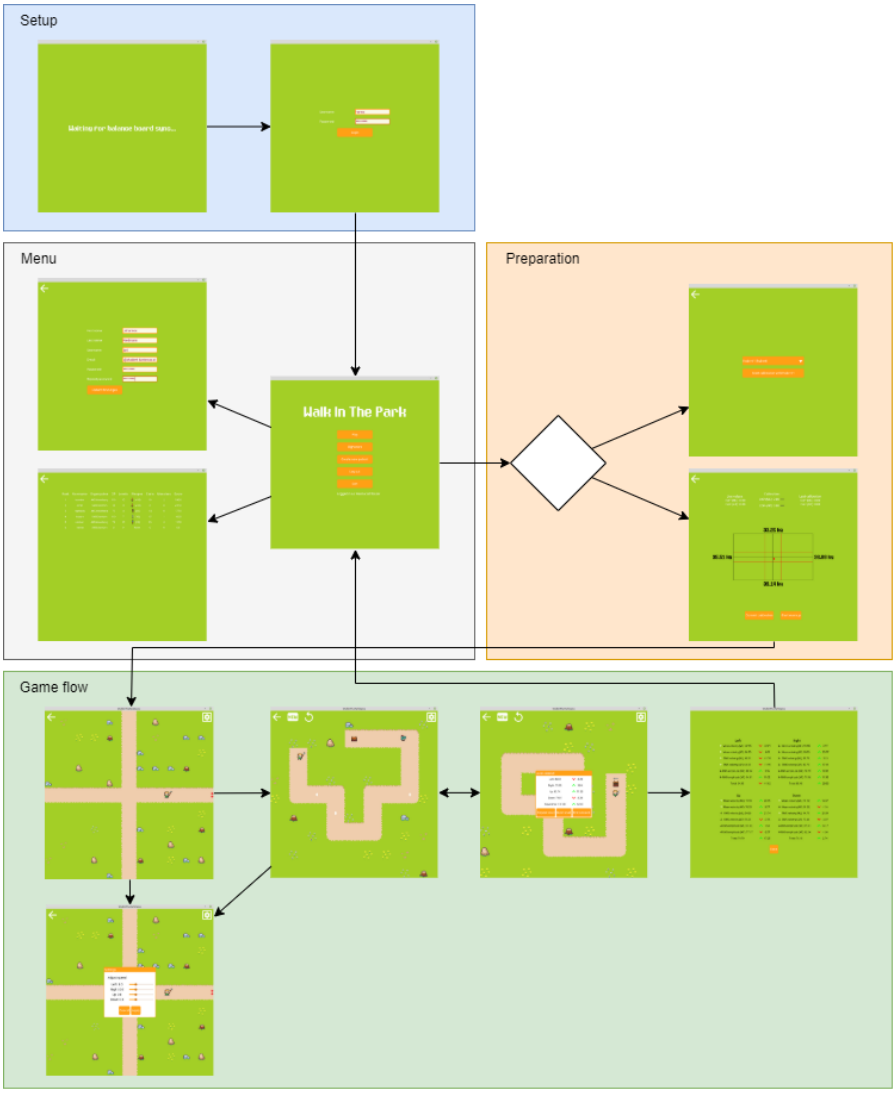


Figure 4.19: Final screen flow of *Walk in the park*

WBB connection screen

Analogous to the design phase, the first screen when starting the application is the *LoadingScreen*, which prompts the user to *sync* the WBB with the client. This wording was chosen because the name of the red button below the WBB that has to be pressed is referred to as the sync button. The initial design, presented in 4.2.2, also included additional instructions to refrain from putting any weight onto the WBB until the connection is successfully established for an initial calibration. This calibration was deemed unnecessary, since unanticipated COP excursions were only present when no weight was applied to balance board.

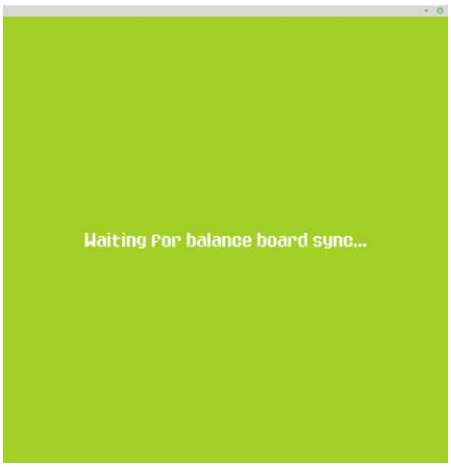


Figure 4.20: WBB connection screen

WiiRemoteJ To establish a stable connection to the WBB and allow data exchange between the WBB and the client application, the library *WiiRemoteJ* [82] (*version 1.6b*) was utilized. WiiRemoteJ depends on an implementation of the JSR082 specification (**J**ava **S**pecification **R**equest regarding Bluetooth for Java). For this purpose, the BlueCove [83] library, which implements the JSR082 specification, was included into the project. The UML class diagram, shown in figure 4.21, illustrates the class interaction to establish the WBB connection. These classes are located in the service package of the application's source code.

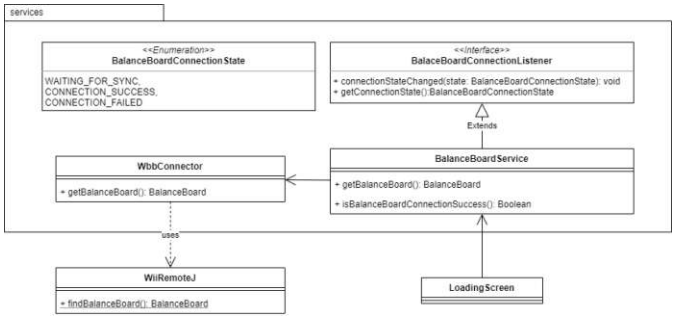


Figure 4.21: Class interaction to establish WBB connection

To summarize, the *LoadingScreen* utilizes an instance of *BalanceBoardService* to acquire the *BalanceBoard*. *BalanceBoardService* is a singleton service that delegates the actual acquisition of the *BalanceBoard* instance to *WbbConnector*. Furthermore, *BalanceBoardService* implements *BalanceBoardConnectionListener*, which gives it the additional responsibility of reporting connection state changes. An instance of *BalanceBoard* is only provided by *WiiRemoteJ*, if a connection to a nearby WBB was established. An outline of this procedure in the form of pseudocode is presented in listing 4.5.

```
BalanceBoard board = null;
balanceBoardConnectionListener.connectionStateChanged(WAITING_FOR_SYNC)
while (board == null) {
    try {
        board = WiiRemoteJ.findBalanceBoard();
        balanceBoardConnectionListener.connectionStateChanged(CONNECTION_SUCCESS)
        return;
    } catch (Exception e) {
        Thread.sleep(3000)
    }
}
```

Listing 4.5: Polling for WBB connection

If *WiiRemoteJ* can not detect a WBB in synchronization state, which means the red button on the backside was pressed recently, an exception is thrown. Every three seconds, an attempt to find a WBB is made, which constitutes a polling mechanism. If a connection to a WBB is established, the connection state of *BalanceBoardService* is modified and a reference to the *BalanceBoard* is kept within the *BalanceBoardService*.

Login screen

To provide user authentication a simple login screen was implemented that reflects the initial design described in section 4.2.2. Physiotherapists and patients both possess credentials to log into the system and start new game sessions. After confirming the provided username and password by clicking the login button shown in figure 4.22, a request is sent to the backend. The login API contract description can be found in table 4.7. The backend responds with either a *User* object if the credentials were valid, or with an HTTP status code 401, which represents an attempt at unauthorized access. In case the entered combination of username and password does not exist, explicit feedback is

provided to the user by a red label under the Login button stating incorrect credentials. Alternatively, in case of a successful login attempt, the user is navigated to the main menu of *Walk in the park*.

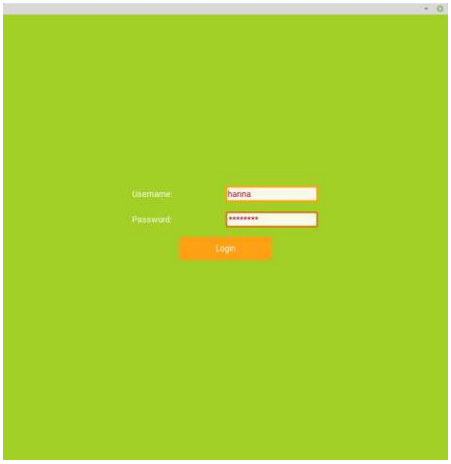


Figure 4.22: Login screen

Menu screen

The main menu is the central point of navigation in *Walk in the park*. The only relevant implementation detail for this screen is when logged in as a therapist, the "Create new patient" button is displayed. This functionality is hidden for patients. Besides that, the menu screen is implemented according to the design presented in section 4.2.2.

Figure 4.23: Main menu

Highscores screen

In comparison to the design, specified in section 4.2.2, the implementation of this screen contains all metrics necessary to compute the final score, as well as the organization the patient is associated with. Listing the organization has the potential of introducing an additional competitive factor to the game. The final score is computed by adding up *XP*, *weapons*, *coins* and *monsters*. As described in the design section, the high score list is not meant to reflect the patient's skill, but rather the time and effort they put into rehabilitation. *Coins* and *monsters* have a chance to appear along the path in each level. The number of coins collected and monsters slain is a reasonable indicator for rehabilitational effort. Weapons can be found in the chest at the end of each level. Their value directly correlates to their rarity, thus the more levels a player completes, the higher the chance they receive a rare weapon. The *XP* value is the only value that correlates with the player's in-game performance. It is also referred to as experience score. The implementation details of experience score computation are described in more detail in section 4.4.5. The high score list also shows the total number of completed *Levels*. This number is not part of the high score calculation, yet it a good indicator of how engaged the patient is with the game.

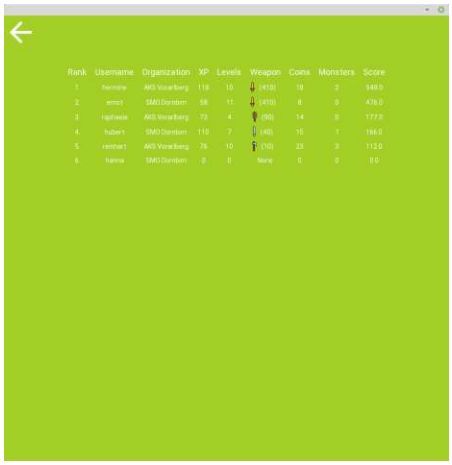


Figure 4.24: Highscores screen

Create patient screen

As a therapist, *Walk in the park* offers the possibility to add new patients to the system. The ramifications of requirements on the design are discussed in section 4.2.2. The input mask contains all fields necessary to fully define a *User* object according to the data model 4.3.3. The entered data is used to construct a *User* object, that is serialized into a valid JSON format and sent to the backend, utilizing the appropriate API endpoint */api/user/*, specified via the API contracts 4.3.4. Specifics about client-server communication are described in section 4.3.2.

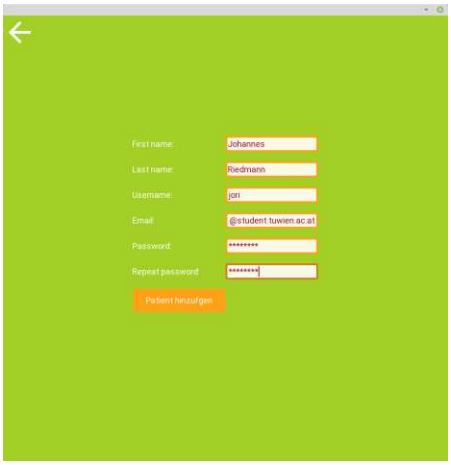
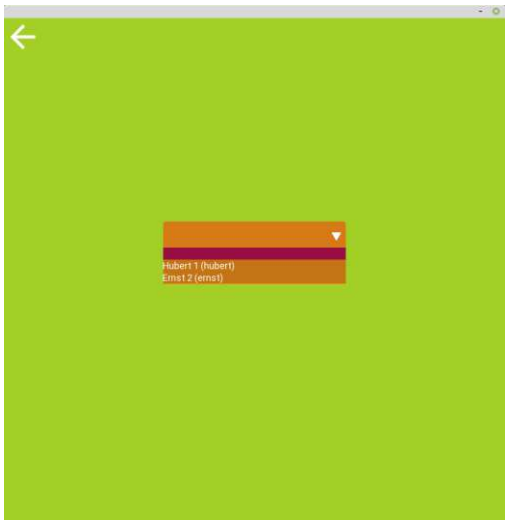


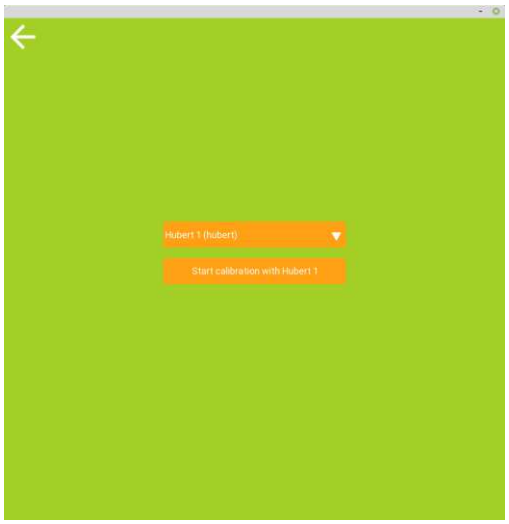
Figure 4.25: Create patient screen

Select patient screen

When logged in as a therapist, this is the first screen when selecting the option to "Play" on the main menu. The reference design of this screen is described in section 4.2.2. When patient selection is opened, the drop-down element is populated by fetching the list of patients that are associated with the logged in therapist's organization. When no patient is selected, a session is assigned to the currently logged in therapist.



(a) List of patients to select



(b) Patient selected

Figure 4.26: Select patient screen

Calibration screen

The rationale behind the calibration screen is described in section 4.2.2. When performing a calibration, the patient is asked place their feet on predefined positions and assume a neutral upright stance. In figure 4.27a, the current COP is visualized as a red dot adjusting to changes in weight distribution at a refresh rate of 20Hz. During development of the prototype, it became apparent that minor fluctuations in COP should not contribute to calibration, thus a threshold was introduced that defines a minimum offset from the coordinate system's center. Optimally, the patient is capable of maintaining the red dot within the predefined calibration threshold of $\pm 20\%$ from the center, visualized by the red lines running perpendicular to the x- and y-axis. This indicates that the patient is not affected by a hemiparesis severe enough to affect their ability to play the game. In case a patient does have a weight bearing asymmetry that exceeds the threshold, the offset from the coordinate system's center (ΔCOP) is assigned to the current *Session* object, see data model in section 4.3.3. ΔCOP is used to shift the center of the original coordinate system to better align with the patient's impairments. Calibration is achieved by interpolating COP measurements during gameplay as defined in equation 4.1.

$$COP_{calibrated} = \begin{cases} COP * (1 + \Delta COP) - \Delta COP, & \text{if } COP > \Delta COP \\ COP * (1 - \Delta COP) - \Delta COP, & \text{otherwise} \end{cases} \tag{4.1}$$

Equation for COP calibration

Figure 4.27a also shows the weight distribution along each axis, which serves as quantification of exhibited asymmetry. Furthermore, on top of the calibration visualization, the current COP values, the committed calibration and the calibration values of the latest session are displayed. The current COP measurements ("Live values"), displayed in the top-left corner, show the current COP measurements in real-time if they exceed the threshold, otherwise zero. Figure 4.27b shows the calibration screen when a new calibration is committed by clicking the "Commit calibration" button. These new values are compared against the calibration associated with the patient's last session, which is fetched from the API. A green arrow pointing upwards shows positive progress, thus a decrease in COP adjustment. Calibration values remaining unchanged are visualized by a gray line and finally, an increase in necessary COP adjustment in depicted as red arrow pointing down. When calibration for the patient is completed, the warmup can be started.

(a) Calibration screen (b) Calibration screen with progress visualization

Figure 4.27: Calibration screen

Warmup screen

An earlier version of the warmup screen, as shown in figure 4.10 in the corresponding design section, shows a more complex path than the final warmup screen, depicted in figure 4.28. This is because the path generation algorithm was used during the design phase to have a reference of discussion about the usefulness of a warmup phase with the physiotherapists. The final warmup screen contains a static level that provides players with a familiar starting point across sessions. The path configuration is defined in *WarmupFloorLayer*. To make the level more visually appealing, different objects are rendered randomly on the background tiles. Attempting to navigate to the previous screen will prompt a dialog warning that the session will end when confirmed. If the player ends the session during warmup, no data will be sent to the backend and the main menu will be shown. When the player is finished warming up, they can start the game by moving towards the red arrows to the right of the screen. If the player has a very limited range of motion, thus only moves very slow in-game, the player velocity can be increased by accessing the game settings.

Figure 4.28: Game warmup screen

Game settings dialog

The game settings provides the option to adjust player velocity independently for each direction. If the player has difficulties to move the in-game avatar due to severe balance impairments, this functionality boosts in-game player velocity without influencing the COP measurements and subsequently not misrepresenting their achieved balance score. Movement can be slowed down by max. 50% or sped up by max. 150% with a step size of 10%.
As shown in the data model in section 4.3.3, the *VelocityAdjustment* is directly associated with *LevelData*, which means that velocity adjustment is stored on a per-level basis. Since the game settings are accessible from both the *WarmupScreen* and the *GameScreen*, fetching the initial velocity adjustment values and handling the dialog is a responsibility of the *BaseGameScreen* class. To retrieve the most recent velocity adjustment for a player, the latest completed *LevelData* object is requested from the backend, utilizing the */api/leveldata* endpoint defined in the API contracts in section 4.3.4. The retrieved *VelocityAdjustment* is then assigned to the *BalanceBoardMovementProvider*, which incorporates the adjustments into the computation of player position based on COP measurements and calibration.

Figure 4.29: Game settings dialog

Game screen

The game screen is the most vital part of *Walk in the park* since it encapsulates the game logic and gamification mechanics that aim to benefit patients with balance impairments on their road to recovery. Functionality bundled into the game screen can be broken down into the following parts: (1) level generation, (2) player movement and recording of COP data, as well as (3) level completion and feedback. These building blocks are discussed in-depth in their respective sections subsequent to this section.

(a) First level (b) Coins and monsters (c) Level cleared

This section focuses on implementation details of the design introduced in section 4.2.2. As described in the design, additional game mechanics and gamification elements can help boost player motivation and engagement. One of these mechanics is that the player can collect coins that are randomly placed along the path, as depicted in figure 4.30b. The number of coins is a randomly generated within the range of 0 and $(numGamePathTiles - 2)/3$, where $numGamePathTiles$ refers to the total number of tiles within a fully specified game path. As mentioned in the design specification in section 4.2.2, to collect a coin, the player does not have to actually touch the coin but

rather just step on the tile that the coin is placed on. Creating incentives to adjust weight distribution during gameplay for the sake of collecting in-game items, or slaying monsters, would be counterintuitive for the underlying goal of the application, which is assessment of balance performance. Thus, the player should not have to move towards in-game objectives, but simply collect them along the way to not distort balance evaluation.
Reaching the chest located on the last tile of each path completes the level and displays their balance evaluation for the level. This is illustrated in figure 4.30c. Furthermore, it rewards the player with either a single gold coin (with a probability of 70%) or a random weapon (30%). Both, the most valuable weapon collected and the total number of coins contribute to the overall score displayed in the highscore list, described in section 4.4.4. There is a total of 14 different types of weapons that can be collected. After completing a level the player inventory, which consists of the *number of monster bones and coins collected* and *a list of collected weapons*, is updated.
Once the player acquires a weapon, there is a chance that monsters spawn and roam along the path. The number of monsters is randomly generated between 0 and $(numGamePathTiles-2)/6$. There is a total of 12 different monsters in the game.
The game screen a player is presented with when playing *Walk in the park* for the very first time is shown in figure 4.30a. From this point on, level generation logic generates personalized levels based on their achieved scores during gameplay and experience playing the game. Further details about level generation and how it implements requirements listed in table 4.4 are discussed in section 4.4.5.

4.4.5 Level generation

In order to boost patient motivation and engagement, *Walk in the park* implements sophisticated level-generation logic that aims to tailor game levels to patient needs. This is achieved by recording COP measurements during gameplay, evaluating balance performance for each direction and subsequently computing an adequate path constellation based on historic and current session performance as well as the player's experience score.

Figure 4.31 outlines the necessary steps for level generation in *Walk in the park*.

Figure 4.31: Level generation flow

As a precondition to level generation, a list of valid path configurations must be computed and written to a file. This is handled by a separate executable, referred to as *PathParamsGenerator*, which is described in section 4.4.5. This section also dives into

details about the necessity of pre-computing valid path configurations.
Level generation is parameterized by two variables: (1) player experience score and (2) overall balance performance for each direction.
When a new session is started, the player's experience score and their average overall balance scores throughout all completed sessions are retrieved from the backend, as described in the following section. The subsequent sections aim to describe how the experience score is utilized to compute a viable level size, matching the player's skill and how the total number of tiles the level encompasses is derived from level size.
Thereafter, an optimal path configuration is computed, converting player balance performance scores to a discrete number of tiles expected for the given level size. The implementation details of this step are also explained in this section.
Finally, this section describes how the path configuration is converted into a renderable path, which adds a start and end node, as well as connecting nodes. The outcome of this step is used to render the in-game tiles to create a personalized level for the player.

Requesting performance data

When the *GameScreen* is first shown, historic performance data across all previously completed game sessions is fetched from the backend. This data is used to compute the mean performance for weight shifts conducted during gameplay in every direction. While interacting with the game, the constant stream of COP values received from the WBB is recorded on a per-tile basis, resulting in a list of tiles associated with COP measurements. Furthermore, since every tile has directional information associated with it, the expected direction for the COP measurements is known. After every completed level, the recorded COP data is sent to the backend for performance evaluation. This request, defined in the API contracts in section 4.3.4, returns an *EvaluationResult*, as described in the data model in section 4.3.3. The *EvaluationResults* for the current session are stored and incorporated into this computation. Combining historic and current balance performance into level generation is specified by requirement **LevGen-2** in table 4.4. Incorporating current session data for each newly generated level also steadily increases level difficulty during a session, which fulfills requirement **LevGen-3**.

Path generation

Path generation is based on a combination of player experience, represented by the player's experience score, and performance broken down to directional weight shifts. Hence, level size, or the length of the generated path, is derived from player experience score, whereas the path configuration, corresponding to the number of tiles for each direction, is inversely proportional to the player's calculated ability to perform weight shifts in the given direction.
To generate levels that respect the boundaries set by a maximum level size, or *gridSize*, and conform to additional constraints aimed to generate more enjoyable levels, a path generation heuristic is utilized. The output of this heuristic is a list of viable path configurations for level generation, stored in a file (*assets/valid_pathgenerator_params.json*).

Thus, the path generation heuristic has to be executed before the application is started. In *Walk in the park*, personalized level generation based on player performance is achieved in the following steps:

(1) Generate a list of viable paths for level generation (precondition, computed once),

(2) Retrieve/Calculate player experience to derive level size and

(3) Compute an optimal path based on historic and current performance data and find the closest match to a viable path.

The following sections describe each of these steps in more detail.

Path generation heuristic In this step, a list of possible path configurations, based on a set of rules, is computed and written to a file within the project resources. Generation of valid paths can be viewed as an independent application, producing a file that the *Walk in the park* can access. During development, this step had to be executed once to produce the necessary file. In case *Walk in the park* needs to be deployed as a bundled artifact, the path generation heuristic has to produce this file first, since level generation depends on it. The pathgen package contains all classes responsible for the path generation heuristic. The resulting path acts as blueprint for the subsequent level rendering. Figure 4.32 outlines the contents of this package.

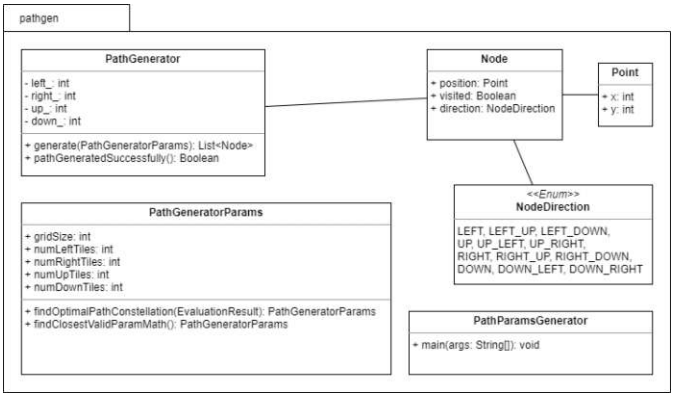


Figure 4.32: Classes responsible for path generation

The *PathGenerator* class contains the heuristic for path generation. This algorithm, more specifically the *generate* method exposed by this class, is parameterized by an instance of *PathGeneratorParams*. This class encapsulates the grid size, which correlates to the total level size and a path configuration, which refers to the amount of tiles for each direction. Figure 4.33 illustrates the smallest possible level size with 4 by 4 tiles, which corresponds to a *gridSize* of 4. In the easiest possible scenario depicted in figure 4.33, 4 by 4 tiles can

be represented by a 2D matrix of *Nodes* to find a suitable path. The grey tile, or node, represents the starting position for the algorithm.

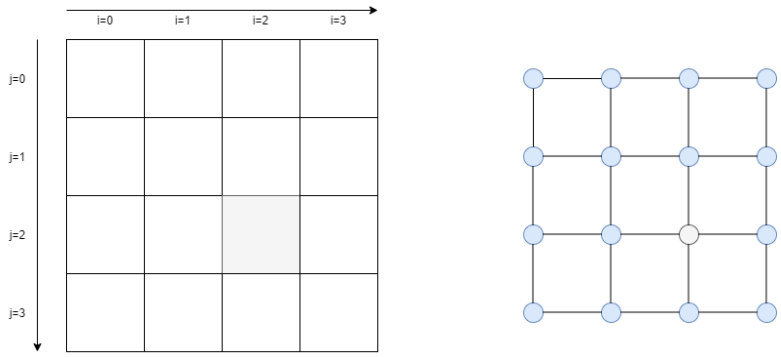


Figure 4.33: 4 by 4 tiles (left) and corresponding grid representation (right) for path generation

The path generation heuristic, described in algorithm A.1, attempts to find a suitable path configuration within a 2D matrix, or grid, of *Nodes*. If no path is found after 100 iterations, the algorithm returns a partial path found up to this point. For each iteration, the initial position is set to approximately the grid's center. A full path is found, if the internal counters of *PathGenerator* for each direction, as described in the class diagram in figure 4.32, match the expected number of *Nodes* for each direction defined by *PathGeneratorParams*. Until this is achieved, a node at the current *position* is picked. The next step is to find a valid *NodeDirection* for this node that contributes to defining a path. This is the core logic for the path generation heuristic. The following conditions have to apply for a *NodeDirection* to be considered as adequate subsequent direction:

(1) **The path should contain exactly the amount of *Nodes* in specific directions as specified in *PathGeneratorParams*.**
For example, if *PathGeneratorParams* has *numLeftTiles* of 2 and *numUpTiles* of 1 and *path* already contains a *Node* assigned to *NodeDirection* UP, the only valid *NodeDirection* for subsequent *Nodes* is LEFT.

(2) **The next *Node* has to be within grid bounds.**
When the last *Node* is on index $i = 0$, as illustrated in figure 4.33, assigning another *Node* to *NodeDirection* LEFT is impossible, since it would leave grid bounds.

(3) **A *Node* can not have more than one direction, thus can only be visited/assigned once by the algorithm.**

(4) **Adjacent *Nodes* are not allowed to have inverse directions, e.g. a LEFT *Node* can only be followed by another LEFT, an UP or a DOWN *Node*.**

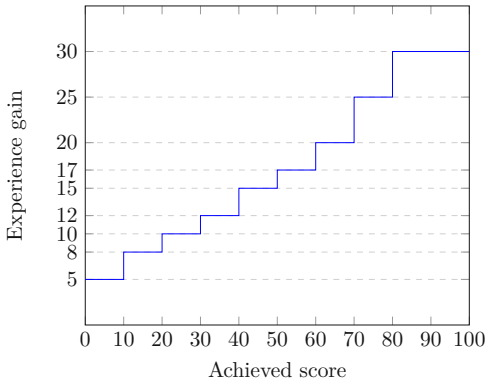
It can not be followed by a RIGHT *Node*.
This constraint prevents lazy path generation, leading to small and confusing levels in which the player has to constantly pace back and forth in horizontal or vertical direction.

A pseudo code implementation of the path generation heuristic is shown in item A.1 in the appendix. The method *getNextDirection(currentNode)*, occuring on line 8 in algorithm A.1, is responsible for evaluating these constraints. If a valid *NodeDirection* was found, it is assigned to *currentNode*. Furthermore, *currentNode* is flagged as *visited*, which marks it as part of the path, thus it can not be added again. Finally, the internal counter for number of tiles for the given direction is decreased. If all internal counters, namely *left_*, *right_*, *up_* and *down_*, match the path configuration specified in *PathGeneratorParams*, the algorithm terminates. In case no *NodeDirection* is found that fulfills the above criteria, path generation is re-attempted. The logic within the *while*-loop, which corresponds to lines 6 to 18 of algorithm A.1, can be considered the path generation heuristic. The surrounding *for*-loop is just a mechanism to re-attempt path generation if the heuristic is exhausted, thus was not able to find a viable direction for a given *Node*, represented by the additional criterion *direction != null* of the *while*-loop. If this heuristic does not terminate within 100 iterations, the algorithm returns a partial path.

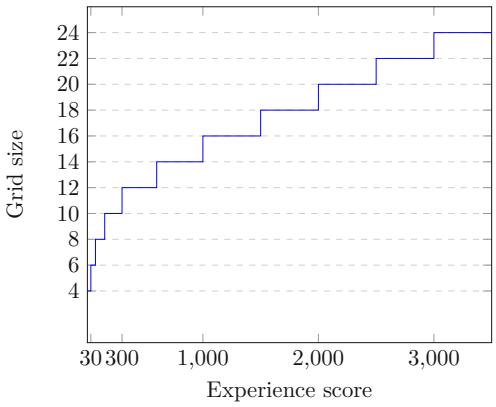
Not every pairing of grid size and path configuration yields a valid path that respects all necessary constraints. Therefore, if the algorithm does not terminate within 100 iterations, the employed *PathGeneratorParams* are considered invalid. Consequentially, a level can not be rendered if the algorithm is parameterized with these values as input. To circumvent this issue, *PathParamsGenerator* was introduced. This class is a separate entry point, thus constitutes an independent application that ultimately generates a file containing valid path configurations for varying grid sizes. This class invokes the *generate* method of *PathGenerator* for a reasonable range of combinations of *PathGeneratorParams*, verifies that the heuristic terminated with a valid path, stores the valid *PathGeneratorParams* and writes them to a file. *PathGeneratorParams* are configured to have a minimum *gridSize* of 4 and a maximum *gridSize* of 24 with an increment of 2. The total number of tiles for levels of a specific *gridSize* is calculated as $totalNumPathTiles = 2 * gridSize - 6$, as shown in line 2 of algorithm A.3 (see appendix). This algorithm is parameterized by a *gridSize* and returns a list of valid *PathGeneratorParams* associated with it. This is achieved by naively iterating over all possible permutations of *PathGeneratorParams*, given the total number of path tiles derived from *gridSize*. If the total number of tiles defined by the current *PathGeneratorParams* configuration matches the total amount of tiles expected for the path, the path generation heuristic is invoked. If this process finishes successfully, the *PathGeneratorParams* are added to the list of valid parameters. The final list contains a total of 12547 valid level configurations. An implementation in pseudo code for the described functionality of *PathGenerator* can be found in the appendix, see item A.1.

Player experience score After completing a level, performance scores are compared to the player's overall performance for each direction. When a relative improvement is achieved, experience is gained, otherwise the player's experience score slightly decreases. An implementation in pseudo code can be found in the appendix, specifically algorithm A.2.

The *computeXp* method returns an experience score gain based on the current score. When *currentScore*. which is the achieved performance score for a given direction, is above or equal to 80, the maximum amount of experience (30 points) is returned, decreasing towards a minimum of 5 points. If the player achieved a total score above their previous average, they gain experience based on the function visualized in figure 4.34a. In case the player scores below their average, the experience score decreases as defined in line 3 of algorithm A.2. Player experience score constitutes the basis for evaluating level size. Figure 4.34b visualizes how level size, or grid size, increases in relation to experience score, starting from the minimum grid size of 4 up to a maximum of 24. The minimum value was chosen based on visual appearance and also because a smaller level would not make sense. The maximum level size was chosen based on visibility constraints. With an increase in level size, textures shrink appropriately due to them being rendered within the shrinking grid. To better support players with visual impairments, a maximum grid size of 24 was deemed viable.



(a) Experience gain returned by *computeXp*



(b) Grid size based on player experience

Compute path configuration To compute a suitable path configuration the total performance measure for weight shifts in each direction, represented by a number between 0 and 100, has to be converted into the expected number of tiles, such that a path/level can be generated. The algorithm for converting an *EvaluationResult* to a optimal path configuration, implemented in pseudo code, can be found in the appendix, item A.4. This piece of code ensures that a path configuration is created, prioritizing tile assignment for the player's weakest direction, corresponding to their lowest weight shift performance result. This ensures that level generation respects requirement **LevGen-1**, defined in table 4.4, but also incorporates weight shifts in other directions to keep the player's motivation and engagement high. Although this algorithm yields an optimal path

configuration based on the player's *EvaluationResult*, the resulting path is not necessarily valid for level generation. As an additional step, to guarantee passing a valid path configuration to level rendering, the best match to the optimal path is looked up in the list of valid path configurations. If there is a perfect match, the optimal path is returned, otherwise the path most closely resembling the optimal path configuration is used.

Conversion to renderable path

To create a list of *Nodes* with associated *NodeDirection* that can be drawn to the screen, the personalized path is augmented with *connecting nodes*. In addition to LEFT, RIGHT, UP and DOWN, the *NodeDirection* enumeration, as illustrated in figure 4.32, also contains values for *Nodes* connecting other *Nodes*, such as LEFT_DOWN or RIGHT_DOWN. Using the level depicted in figure 4.35b as an example, the optimal personalized path configuration, represented by a list of *Node* objects, only contains *Nodes* with directions LEFT, RIGHT, UP and DOWN. Figure 4.35a visualizes the optimal path configuration for the level depicted in figure 4.35b. Every circle shown in figure 4.35a can be considered a *connecting tile*, whereas the arrows constitute *direction tiles*. Therefore, before rendering a game level, the optimal path configuration has to be extended with *connecting nodes*. The *connection nodes* are placed where directional changes occur. The list of *Nodes* resulted from optimal path generation for the level depicted in figure 4.35b can be formalized as follows:

D → **R** → **U** → **U** → **U** → **L** → **D** → **L** → **D** → **L** → **U** → **L** → **U** → **U**

Adding the *connecting nodes* results in:

D → **DR** → R → **RU** → U → U → U → **UL** → L → **LD** → D → **DL** → L → **LD** → D → **DL** → L → **LU** → U → **UL** → L → U → U

Furthermore, to complete the path for level generation, a start and end node have to be added. The start node represents the player's spawn location. The player is initially positioned in the center of the starting tile. Only after leaving the starting tile, COP measurements are recorded, hence the start node does not contribute to balance performance evaluation. Another reason for excluding the starting tile from contributing to player performance assessment is to reduce the impact of initial confusion when entering a new level. The player might not be fully prepared right after loading into a new level. Measurements recorded during this very brief initial familiarization phase potentially contribute to misrepresentation of their balance performance. The end node contains the treasure chest, as described in the game concept in section 4.2.1, and is positioned in the end tile's center. Thus, the end tile is also exempt from performance evaluation. Finally, after adding the start and end node, the full path (*game path*) that describes the level depicted in figure 4.35b:

D (*start node*, *player spawn*) → D → DR → R → RU → U → U → U → UL → L → LD → D → DL → L → LD → D → DL → L → LU → U → UL → L → U → U → U (*end node*, *chest*)

(a) Optimal path (b) Level with crossing (c) After leaving crossing tile

Figure 4.35: Level rendered for an optimal path generated for *gridSize* of 6

After a renderable path is fully specified, the path is measured and drawn. For some path constellations, the addition of *connecting nodes* can result in overlapping tiles, or crossings, as illustrated in figure 4.35b. To prevent confusion, red arrows point in the direction the player is expected to move. After leaving the crossing tile, the red arrows point in the direction the player has to navigate the next time they are on the tile, as shown in figure 4.35c.

4.4.6 COP data recording and player movement

This section details how COP data is received, stored and how it is utilized to compute player position within the game levels. This is a vital part of *Walk in the park* since level generation is based on player performance, which can only be accurately estimated when the provided data is sampled in a well-defined manner.

COP data recording

The *WiiRemoteJ* library, described in section 4.4.4, provides the abstract class *BalanceBoardAdapter*, which exposes functionality that enables receiving the sensory data stream emitted by the WBB at 100 measurements per second. This data contains the detected mass in kilograms for each of the WBB's four force transducers. *BalanceBoardMovementProvider* inherits from *BalanceBoardAdapter* and converts the received measurements into COP values, utilizing the COP equations described in section 2.6.1. Every second COP measurement is recorded, sampling COP data at effectively 50Hz. The data model, presented in section 4.3.3, shows that this data is stored on a per-tile basis. The current game level, represented by a *LevelData* object, contains a list of *LevelTiles*. Furthermore, every *LevelTile* has a list of *CopSamples*. A *CopSample* constitutes a single measurement from the WBB's sensory data stream recorded during gameplay. When navigating through a game level, the player passes over the *LevelTiles*. When entering a new *LevelTile*, the new tile is added to the list of tiles in *LevelData* and subsequent measurements are associated with the most recent tile. This allows recording COP data

on a per-tile basis, always keeping track of the expected direction for the performed weight shift during gameplay due to the direction property associated with each *LevelTile*.

Calculating player movement

An additional responsibility of *BalanceBoardMovementProvider* is to update the player position by incorporating calibration and velocity adjustment into the current COP measurement. For COP recording, only raw COP values are used. This allows impaired patients to play the game without misrepresenting their balance performance. When a session is started, the *VelocityAdjustment* associated with the latest *LevelData* object, see data model in section 4.3.3, persisted for the patient is requested from the backend. Calibration values, represented by *CalibrationData*, are associated with a *Session* to track progression over time. Figure 4.36 visualizes the usage of the *copXDiff* and *copYDiff* properties of *CalibrationData*. The idea behind the employed calibration mechanism is explained in more detail in section 4.4.4. Both calibration and velocity adjustment impact how COP measurements are translated into in-game movement.
Figure 4.36 shows how shifting the coordinate system's center *c* to *c'* skews the range of COP values. When linearly interpolating COP measurements to respect the calibrated value ranges, in-game movement towards the player's weaker side becomes significantly easier. As an additional measure, velocity adjustment aims to boost or slow down movement speed into a specific direction by simply multiplying the calibrated COP values with the respective velocity adjustment. Both of these measures aim to support players with severe balance impairments and make the game more accessible.

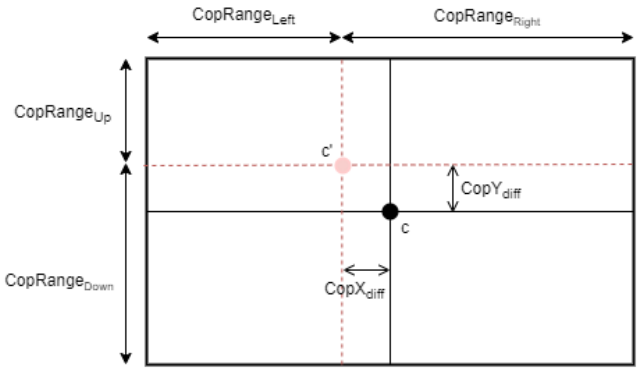


Figure 4.36: Adjusted coordinate system by incorporating *CalibrationData*

Before updating player position, certain constraints have to be checked, since the player is not allowed to move freely in *Walk in the park*.

Movement constraints

Movement is restricted to the path tiles only to simplify recording of COP data. If the player was able to leave the path, association between the current expected weigh shift

direction and the recorded COP data would be impossible. Furthermore, only forward movement along the path is allowed. Once the player leaves a tile, they are not allowed to re-visit it. This simplifies sampling logic and provides a clear timeline for when movement on a specific tile was completed.
Additionally, as discussed in section 4.4.5, COP measurements for the starting and end tile of a path are not recorded. This aims to circumvent a prevalent issue mentioned in literature, which refers to truncating COP measurements to prevent including unrepresentative data. At the start of an exercise, the player might not have analyzed the level properly to instantly shift their weight into the correct direction, resulting in inaccurate measurements at the start of each level. There are two mechanics in place to circumvent this issue in *Walk in the park*: (1) explicitly create a starting tile where the COP is not recorded and (2) introduce a count down at the start of each level.

4.4.7 Level completion and feedback

To provide the physiotherapists with objective data about the patient's balance performance, after every level and at the end of each session, performance scores are presented.

Level feedback

Design considerations and requirements incorporated for this screen are discussed in section 4.2.2. As an extension to the design, figure 4.37 includes progress in experience score for the player. The concept of an experience score was added as implementation detail to parameterize level generation, as discussed in more detail in section 4.4.5, thus it was not present at the time of mockup creation. Furthermore, the final implementation of the level feedback dialog does not contain a total score for a single level, because the average result across all directions does not contribute to understanding balance performance.
After reaching the treasure chest on the ending tile, as shown in figure 4.37, the current *LevelData* and associated *LevelTiles*, containing the COP measurements, are sent to the backend for score evaluation. To achieve this, the *LevelData* object is sent to the backend using the */api/session/<sessionId>/leveldata* endpoint, see API contract in section 4.3.4. As a result, this endpoint returns an *EvaluationResult*, see data model in section 4.3.3, which contains a *ModelResult* for each direction. *ModelResult* encapsulates evaluation results for each model type. Level feedback is kept concise to not overwhelm players and physiotherapists with information, thus only the total score for each *ModelResult* is displayed, disregarding results for each distinct parameter.
After storing *LevelData* and retrieving *EvaluationResult*, the received treasure is added to the player's inventory, their experience score is computed and finally, the updated *User* information is sent to the backend utilizing the */api/user* endpoint.

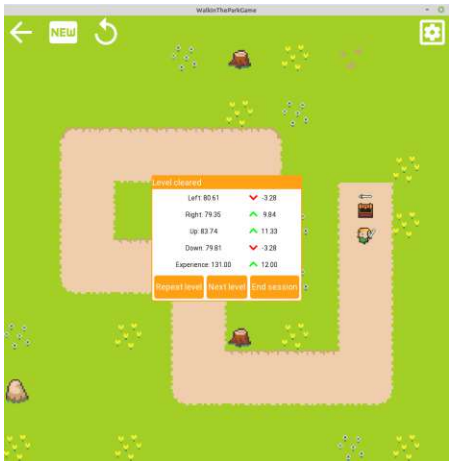


Figure 4.37: Level feedback

Besides showing the patient's balance performance and visualizing their progress, this screen allows to re-play the current level, proceed to a new level or end the current session.

Session feedback

When the player decides to end the current session and at least one level was completed during the session, the API endpoint */api/session/<sessionId>/end* is called. This request results in an updated *Session* object, which contains an *EvaluationResult* for the whole session. If the player ends a session without completing a level, they are navigated to the main menu.
Implementation of the session summary screen is based on the design discussed in section 4.2.2. Figure 4.38 shows the rich feedback received in the context of a session summary. This is vastly different from the original design, because at the point of designing the mockups, it was not clear which parameters or machine learning models are going to be used for balance evaluation.
The session summary presents the evaluation results per direction. For each direction, COP-based parameters are listed, as well as a total score, which directly represents a *ModelResult* object (see the data model in section 4.3.3). The white upwards arrow visualizes that the parameter was overstepped in relation to model data, whereas a black downwards arrow refers to the value being below the healthy baseline data. Analyzing these values on a per-parameter basis can yield valuable insight into the balance behavior of the patient. Understanding characteristics of these parameters is vital to reading the session summary. A more detailed discussion about the expressiveness of these parameters is presented in section 2.6.2.
As opposed to the parameter-based evaluation results, the total result represents a more sophisticated balance measure. Instead of deriving insight from a one-dimensional distribution, the total score is derived from a multivariate normal distribution, which

incorporates correlations between the parameters. This is discussed in greater detail in section 4.6.2.
Each evaluation metric visualizes progress in comparison with overall historic player performance. The green upwards arrow shows positive progress, which corresponds to getting closer to balance behavior exhibited by healthy individuals, whereas a red downwards arrow indicates the opposite.

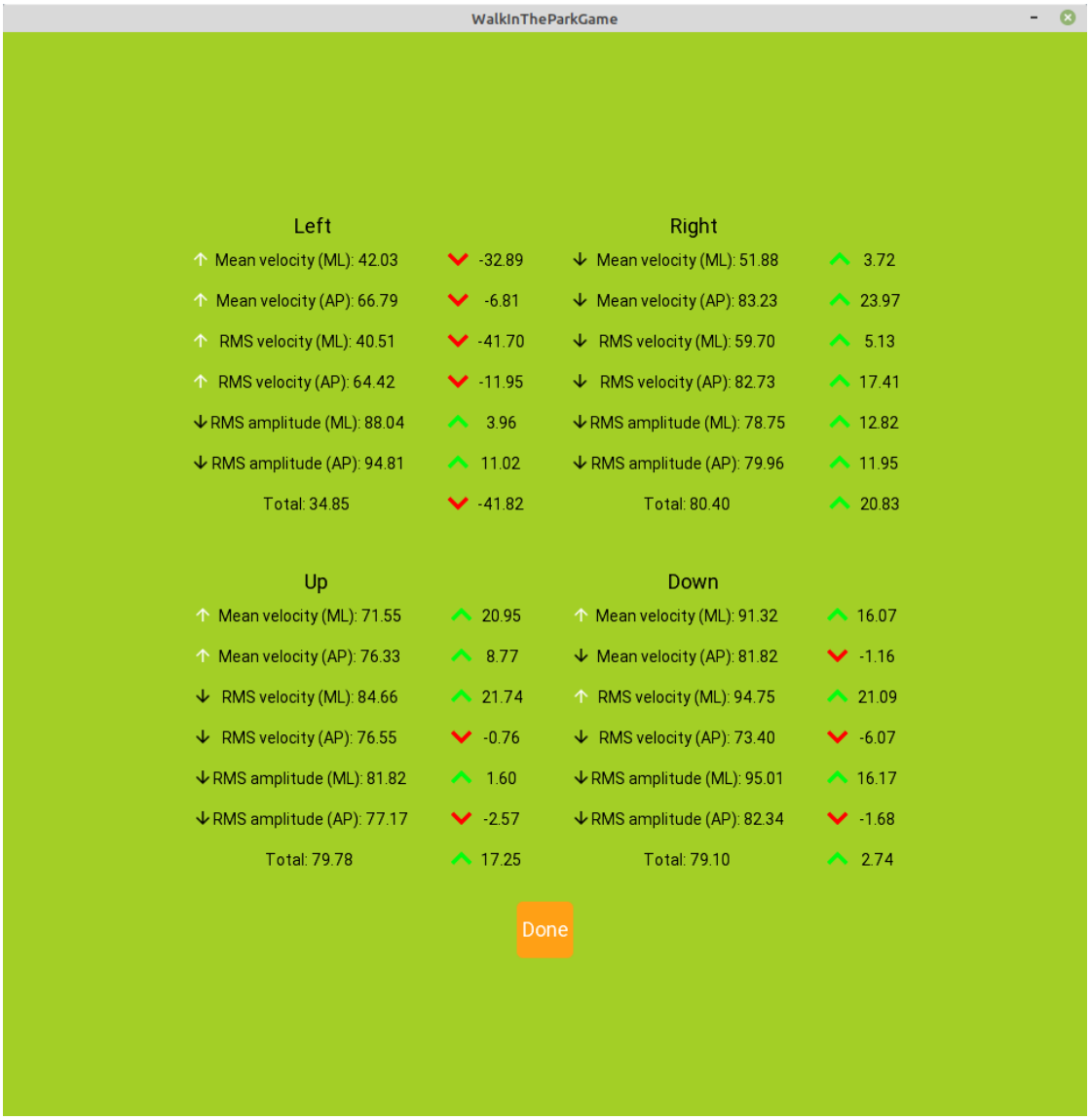


Figure 4.38: Session summary

4.5 Walk in the park - Backend

Walk in the park's backend is implemented as a Spring Boot application, which offers convenient abstractions to accelerate development. Section 4.1.2 provides further information about the technology choice. When creating a webservice with Spring Boot, classes marked with the *RestController* annotation are used to handle HTTP requests. In case of *Walk in the park*, these are the *LoginController*, *UserController*, *SessionController*, *LevelDataController* and *HighscoreController*. Figure 4.39 shows how the controllers fit into the backend's architecture. They form the *business layer* of the application. *RestControllers* map specific URLs that uniquely identify a *resource* to Java methods. Section 4.3.2 provides a more detailed discussion about the role of *resources* in client-server communication. The mapping of a request URL to a Java method defines which piece of code is executed when receiving an HTTP request. The API contracts, defined in section 4.3.4, specify the expected behavior of these methods.

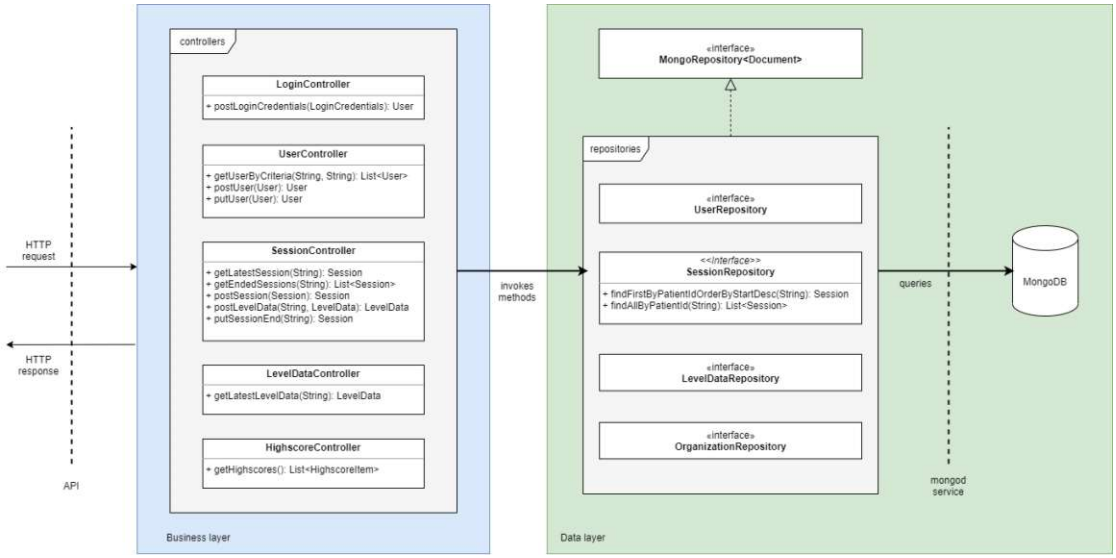


Figure 4.39: Backend software architecture and packages

The two main responsibilities for the backend of *Walk in the park* are (1) providing data and (2) storing data. The following sections will outline the implementation details of how it handles these responsibilities.

4.5.1 Data access

Spring Boot offers numerous powerful integrations out-of-the-box. For the *Walk in the park* backend, the *spring-boot-starter-data-mongodb* package was particularly useful, because it provides easy access to data within a *MongoDB* document database instance. Repositories for specific document types can be created by simply extending *MongoRepository* provided by this package. All of the repositories created for *Walk in the park*, illustrated in figure

4.39, representing the *data layer*, extend *MongoRepository*.
When the backend application is initialized, Spring Boot creates instances of these repositories, thus they become available within the whole application. To access the database, every *RestController* utilizes the necessary repositories to serve requested data in a well-defined manner to the frontend. Communication between repositories and the database is based on the *mongod* service, which "*is the primary daemon process for the MongoDB system. It handles data requests, manages data access, and performs background management operations.*" [84]
The methods exposed by the *RestControllers* represent an implementation of the API contract defined in section 4.3.4. Because all repositories inherit from *MongoRepository*, as shown in figure 4.39, they already implement CRUD operations, thus most of the required specifications for data retrieval are provided out of the box. Solely *SessionRepository* contains two additional method definitions for fetching the latest session for a player and retrieving all completed sessions, representing sessions that contain an end date, for a specific patient.

4.5.2 Data storage

As opposed to relational databases, which strictly adhere to a predefined data model, *MongoDB* is a document-based database, which means that it stores all data as JSON-like documents. Documents are stored in so-called *collections*, which represent separate storage units for each type of document. The *Walk in the park MongoDB* instance contains four types of documents, thus four *collections* in total. The repositories, depicted in figure 4.39, reflect these document types, namely *User*, *Session*, *LevelData* and *Organization*. The remaining information, presented in the data model in figure 4.16, is embedded in these four types of documents.
The *MongoDB* instance is accessed by the Spring Boot application to serve data to the *Walk in the park* client and from Python scripts, used to fetch training data for model generation and evaluate level performance.

4.6 Walk in the park - Modeling balance ability

This section describes the steps for modeling balance ability, utilizing COP data of healthy individuals. Firstly, the overall concept is briefly explained, followed by a more detailed discussion of each step towards a model for healthy balance ability, based on the *model lifecycle*, discussed in section 2.7.2. Finally, conversion of model output to individual balance metrics and overall balance score is discussed in detail.

4.6.1 Concept

The contributions in literature, described in section 3.3, follow supervised learning approaches by labeling their training data. A brief description of supervised learning can be found in section 2.7.1. As a consequence of the supervised learning approach, domain experts have to be closely involved when creating the training data and invest a considerable amount of time to label each observation or trial within the training data. This is not a feasible approach for the prototype, because of resource constraints.
To circumvent this issue, unsupervised learning is utilized, which is briefly described in section 2.7.1. Instead of sampling data from patients with balance impairments, quantifying the level of impairment by specialists and deriving a model based on the resulting information, the approach used in this thesis utilizes data from healthy individuals, which greatly simplifies the process of data acquisition. Subsequently, this data is used to model balance behavior for a healthy baseline, utilizing parameters to quantify balance ability taken from literature. Section 2.6.2 provides further insight into these parameters. Instead of quantifying balance ability as a result of supervised classifiers, an unsupervised approach in combination with distance measures is applied to calculate the deviation of new observations in relation to the healthy model. In other words, a score is calculated by comparing new measurements of impaired patients, taken during trial, to a model representing balance behavior of healthy individuals.

4.6.2 Building the model

The following sections describe how healthy balance ability was modeled in the scope of this thesis, aligning each step with the machine learning pipeline, discussed in section 2.7.2.

Data acquisition

The first step of building a machine learning model, as described in the section about the machine learning pipeline 2.7.2, is to acquire data. The goal of data acquisition in *Walk in the park* is to collect sufficient information to derive indicators that capture balance behavior of non-impaired individuals. To achieve this, data of 11 healthy volunteers was collected over 34 individual sessions. Participants for this phase were between 26 and 60 years old, with the average age being 32. Every participant received the same

instructions about how to play the game, how to position their feet correctly and how to perform the weight shifting exercises. During this phase, 219 levels were completed. While collecting training data, the backend was configured to store all data in a separate *MongoDB* instance named *training*, referring to the training data for model generation.

Data ingestion

The tile-based level design has the inherent benefit of automatically associating sequences of COP measurements with their intended direction, referring to the direction of the weight shift. For more details about directional association of tiles, see the game concept in section 4.2.1. This enables grouping COP sequences, corresponding to weight shifts, by direction and analyzing these groups independently. As described in section 4.4.5, a path consists of directional tiles (left, right, up, down) and connecting tiles. In this step, COP measurements associated with connecting tiles are removed, since we are only interested in unidirectional weight shifts. The output of this step is COP measurements, grouped by directional assignment.

Data validation

In this phase, significant statistical outliers are removed from the dataset. Execution of excercises for the collection of training data was observed thoroughly and data was removed as soon as improper execution occurred.
As stated in section 2.7.2, feature engineering is performed as part of this step. In this thesis, features were consolidated by extensive literature research and comparison with state-of-the-art COP-based contributions for balance rehabilitation. A detailed discussion about the considered features can be found in section 2.6.2. Based on these findings, the final set of features for model generation include the COP velocity, RMS amplitude and RMS velocity. All of these features consist of medio-lateral (ML) and anterior-posterior (AP) components. Decomposition into these components enables a more in-depth analysis of balance behavior for each weight-shift direction and results in the following six final features for model training: (1) COP velocity ML, (2) COP velocity AP, (3) RMS amplitude ML, (4) RMS amplitude AP, (5) RMS velocity ML and (6) RMS velocity AP.

Data preprocessing

Due to clear instructions on how to perform the weight shifts and correcting foot positioning and posture during training data collection, no additional processing steps were needed before further data analysis for model generation. As stated in the data acquisition section, training data was collected from 34 sessions, completing 219 levels in the process. Feature extraction is performed on a per-tile basis. The 219 completed levels encompass 13284 tiles, hence this equates to the total number of features extracted. The distribution of tiles across directions is sufficiently even with 3228 left, 2976 right, 3648 upwards and 3432 downwards weight shifts. The following section describes how features

extracted from training data are analyzed and converted into a model, representing weight shifts of an individual with healthy balance ability.

Model training

As discussed in the concept, see section 4.6.1, balance ability is modeled as a form of one-class classifier. This family of models and how it applies to the problem domain of this thesis is described in more detail in section 2.7.3. To derive a suitable model for the training data, the underlying distribution must be analyzed. Figure 4.40 exemplifies the distribution of training data for weight shifts to the left direction for all extracted features. Distributions of training data for right, up and down weight shifts are closely aligned and show comparable properties.

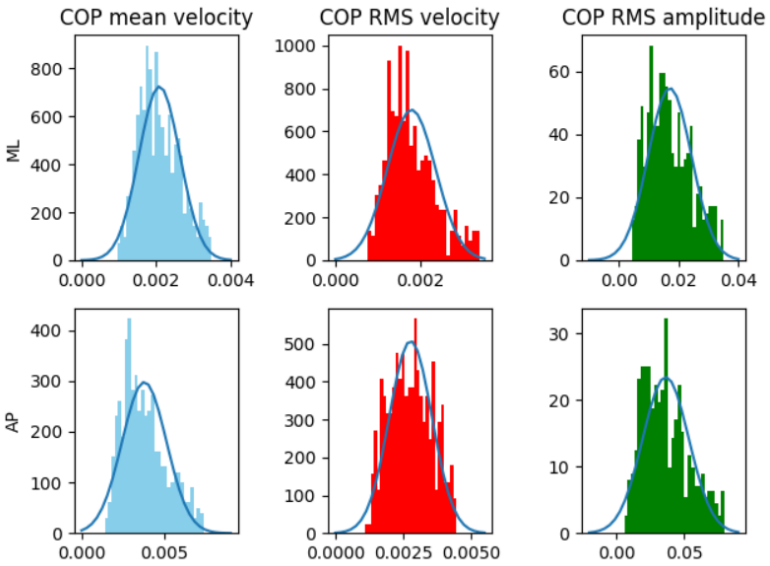


Figure 4.40: Histogram of all features, based on training data for weight shifts in left direction

The data shows that each feature can be approximated by a normal distribution (ND). The corresponding NDs are plotted alongside the histograms. None of the features perfectly fit their associated ND, but they are deemed a sufficient generalization of the underlying training data, see model analysis in section 4.6.2. The resulting one-dimensional (or univariate) ND models do not have to perfectly reflect the training data, but rather approximate the underlying distribution. Each distribution represents a relevant property in healthy balance ability during a weight shift exercise. The underlying assumption is that the more a new measurement deviates from the distribution, the less it represents healthy balance behavior. This allows in-depth analysis of a person's balance ability, since it decomposes their directional weight shifting performance into single parameters that can be utilized to make deductions. The underlying rationale for each feature and

its impact on a person's balance ability is discussed in more detail in section 2.6.2.
A normal distribution, or Gaussian distribution, is defined by its probability density function, see equation 4.2.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{4.2}$$

Probability density function of a ND

To calculate the probability of a new measurement x belonging to a ND, the mean μ and standard deviation σ have to be known. Even though evaluating each feature independently provides valuable insight into the individual components of balance ability, it does not convey correlations between the features, thus lacks the ability to depict an overall balance evaluation. For example, when performing weight shifts to the left or right, for healthy individuals, it is expected that the relation between medio-lateral COP velocity and anterior-posterior COP velocity lies within a certain range. In other words, the correlation between COP velocity ML and its AP component have to be modeled in order to accurately model COP velocity. Furthermore, a model capturing relations between each of the individual features is necessary in order to model healthy balance ability in its entirety.
To achieve this, a multivariate normal distribution (MVND) is utilized, combining the univariate normal distributions into a MVND, resulting in one MVND model for each direction. A MVND can be described as a generalization of the univariate ND to higher dimensions. In case of a univariate ND it is sufficient to store the mean μ and standard deviation σ to quantify any future measurement x belonging to the distribution. This is insufficient for parameterizing a MVND. Equation 4.3 describes the probability density function of a MVND. The parameter k reflects the number of dimensions, or distributions, the MVND incorporates. Since the model entails six univariate NDs, the value of this parameter is six for the purpose of building a MVND for each direction. Furthermore, the covariance matrix Σ must be known to describe the probability density of a MVND. The covariance matrix is a symmetrical, positive semi-definite square matrix, denoting the covariance between each pair of elements of a given vector. Hence, it describes correlations between all six features in six-by-six-dimensional matrix, which enables the MVND to capture relations between each feature.

$$f(x) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{4.3}$$

Probability density function of a MVND

In the scope of this thesis, the relevant part about the MVND is the Mahalanobis distance, see equation 4.4. The Mahalanobis distance is an integral part of the MVND. It

constitutes the distance measure between a new observation x and a given distribution.

$$D_m(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{4.4}$$

Definition of the Mahalanobis distance

In this thesis, utilizing the Mahalanobis distance is essential to comparing new observations with the underlying MVND. The necessary parameters for describing the MVND of any direction is a 1-by-6 vector, representing the mean values for all features, constituting the center of the distribution, and the covariance matrix Σ, describing interrelation between features. Furthermore, for scoring purposes, the Mahalanobis distance for each individual training data point is computed. Section 4.6.3 describes how the mean and standard deviation of the Mahalanobis distances are relevant for the final overall balance score computation.

Model analysis

A significant part of model analysis is validating accuracy. In supervised learning, a test set is used to measure classification accuracy. The test set is a labeled set of observations that are not contained in the training set. In simple terms, the more observations of the test set are classified correctly by the model, the higher the model's accuracy. An unsupervised learning model, specifically a model used for anomaly detection, can be evaluated in a similar fashion. Utilizing a labeled test set, observations can be classified as either inliers or outliers. The more observations in the test set are classified correctly, the higher the model accuracy.
In this thesis, the goal is not to distinguish between inliers and outliers, hence merely detecting healthy and unhealthy balance ability by defining a decision boundary. Instead of a binary decision from the model, new observations are assigned with a probability describing association with the model's underlying distribution. To improve intuition of model output, this resulting probability is converted to a balance score between 0 and 100.
Since balance scores are computed by applying a distance measure to the distribution and a new observation, e.g. Mahalanobis distance for the MVND, the underlying distribution must be validated to guarantee balance scores are calculated accurately. As described in the previous section, a MVND model is utilized to describe the overall balance ability for each direction. Validation of the MVND model can be achieved by comparing the distribution of resulting balance scores with one achieved by training a higher-level anomaly detection model.
To achieve this, an autoencoder was trained on the same training data as the MVND model. An autoencoder is a type of artificial neural network that is capable of learning an encoding, a compressed representation of the training data. The basic schema of an autoencoder shown in figure 4.41.

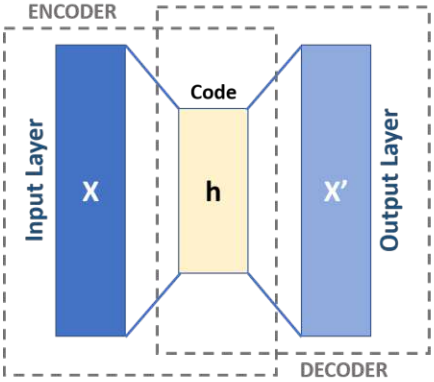


Figure 4.41: Schema of an autoencoder [85]

It consists of an *encoder*, mapping an input X to an encoding h and a *decoder*, mapping an encoding h to a reconstruction X' of the input. The autoencoder, used to validate the MVND model, has six nodes in the input and reconstruction layer, representing the features to describe balance ability and three nodes for the encoded representation h. When training the autoencoder, features of the training set are extracted and fed into the input layer. The autoencoder will learn a lower-dimension encoding h, only keeping the most relevant features, while minimizing reconstruction error. This reconstruction error is how the autoencoder can be utilized for anomaly detection. After learning an encoding from training data, providing the autoencoder with a new observation will result in a reconstruction error. The higher this error, the more the observation deviates from the learned encoding, thus the more anomalous it is. This reconstruction error can be used as a metric to quantify deviation from the model, hence provide the means to compute a balance score approximation. Figure 4.42 visualizes the balance scores for all levels in the training data for left-sided weight shifts, comparing MVND and autoencoder scores.

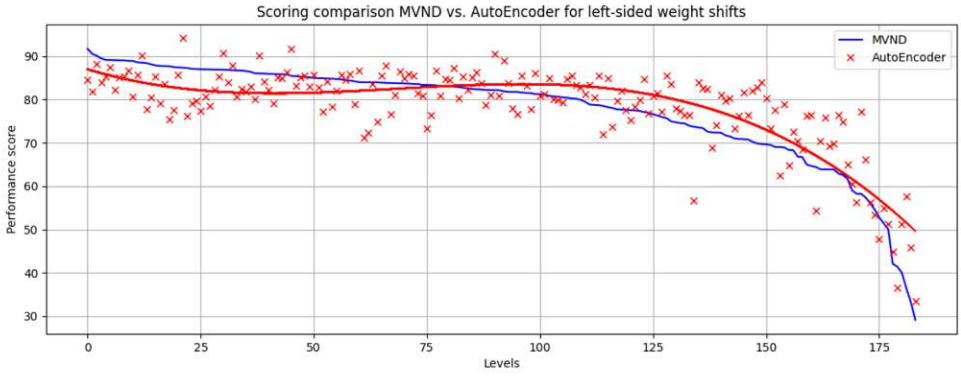


Figure 4.42: Scoring comparison MVND vs. AutoEncoder

The blue line shows the balance score for left-sided weight shifts, sorted in decreasing order, for each level in the training data (levels are indicated by their index on the x-axis). The red x symbols visualize the individual balance scores computed using the autoencoder for the same levels. The red line indicates the statistical trend of the autoencoder scores, utilizing a cubic polynomial fit approximation. Even though the individual autoencoder balance scores deviate from the MVND by $\sigma = 3.45$ points, the overall trend of both balance score estimations are similar. This indicates that the MVND model constitutes an appropriate generalization of the underlying training data, thus a valid approximation for an overall balance score. Section 4.6.3 provides further insight on how the balance score is calculated based on the MVND model output. The autoencoder was not considered as replacement or additional input for balance score calculation because of time constraints regarding inference. Calculating balance performance scores for levels could take up to 10 seconds for levels with 20 tiles or less on the main development machine, mentioned in section 4.3.5. Performing inference on the MVND model does not impose these runtime constraints. Furthermore, based on the undertaken model analysis, the MVND model constitutes an equivalent approximation for balance scoring.

Model deployment

To make models accessible for balance score calculation, triggered by receiving new level data on the backend-side, model parameters must be stored in the MongoDB instance. The model parameters μ and σ are associated with their respective directions and model type and persisted to a new collection within the MongoDB instance, resulting in six univariate ND models per direction and a total of 24 models. Figure 4.43 shows how univariate models are persisted within the MongoDB instance's *Model* collection.


```
_id: ObjectId("5e15e07f43da3e549def3daf")
direction: "LEFT"
type: "cop_vel_mean_ml"
mean: 0.00229494215701418
std: 0.0009021312769377229

_id: ObjectId("5e15e07f43da3e549def3db0")
direction: "LEFT"
type: "cop_vel_mean_ap"
mean: 0.004172837536327178
std: 0.0020122856532032145
```


Figure 4.43: Example showing how univariate ND are stored in the MongoDB instance

For MVND models, in addition to the associated direction and model type, each MVND

model is defined by the 1-by-6 array of mean values, a serialized representation of the covariance matrix, as well as the mean and standard deviation of Mahalanobis distances within the training set. Figure 4.44 exemplifies how the values are stored within the MongoDB instance for the left direction.

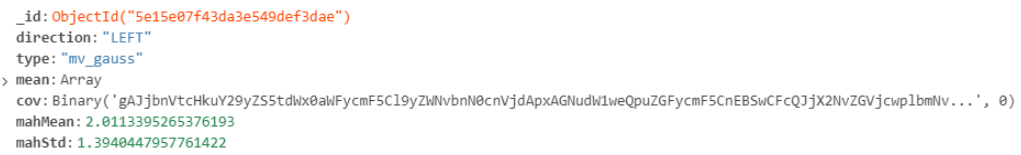


Figure 4.44: Example showing how a MVND is stored in the MongoDB instance

Model feedback

In this thesis, model feedback is gathered during an evaluation phase after completion of the prototypical implementation, based on feedback by professional physiotherapists. According to the distributions of performance scores, shown in figure 4.42, 110 individual levels have been completed with scores of 80 and above. Levels with lower scores are associated with the first few levels the participants played. After session completion, nearly all healthy participants achieved scores of 80 and above for their weight shifts, averaging performance throughout the session. Thus, a threshold of 80 points to distinguish between healthy and impaired balance behavior is deemed reasonable. Further discussion about this threshold can be found in session 5.1.1 as part of the evaluation phase.
Accurately measuring healthy balance ability is not the aim of this prototype. The ultimate goal is to detect weaknesses in players with balance disabilities, thus investigating if the underlying model and balance score computation is able to capture impaired player's largest deficiencies. To evaluate if the model and distance-based balance score calculation can also be applied to players with impaired balance ability, an evaluation was carried out and described in section 5.

4.6.3 Balance score calculation

The balance score is a value in the range of 0 to 100 that aims to capture balance ability during weight shifts, make performance and progression quantifiable and provide vital information about the player's balance deficits. An overview of the flow from completing a level to receiving balance scoring is visualized in figure 4.45. After completing a level, the data gathered during gameplay is sent to the backend. The COP measurements are stored in the *tiles* array, which consists of an ordered list of tiles, representing the completed level. Each tile contains a sequence of COP measurements, referred to as *samples* in the data model, which reflect the player movements on the WBB. For a complete overview of the *LevelData* object and its properties, see data model shown in figure 4.16.
The newly received *LevelData* object is persisted to the database and balance score calculation is triggered. The following sections describe how (1) individual balance metric

scores and (2) balance scores that reflect overall balance ability are computed. The resulting scores are written into *LevelData*'s *evaluationResult* property. After the process performing balance score calculation is finished, the updated *LevelData* object, containing the results of balance score calculation in its *evaluationResult* property, is returned.

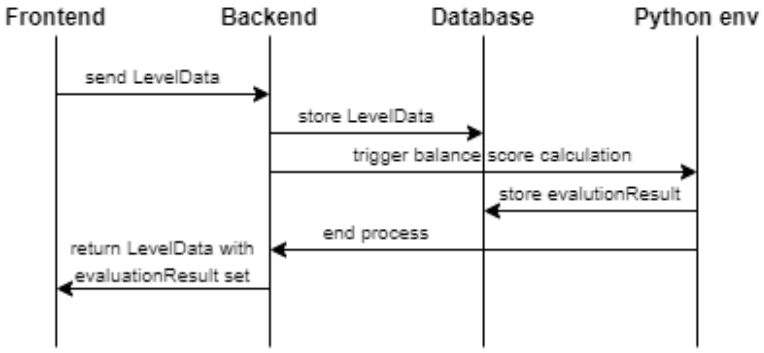


Figure 4.45: *LevelData* flow for balance score calculation

Individual balance metrics

The intuition behind individual balance metrics is to evaluate each extracted feature individually by computing a score that reflects how closely it resembles the healthy baseline. Furthermore, for each individual balance metric, the deviation from the optimal score is calculated. This makes it possible to understand if the mean of the feature's underlying univariate ND was over- or under-stepped. This is vital in understanding the weaknesses in a player balance ability while performing weight shifts. An in-depth analysis of the balance metrics considered in this thesis can be found in section 2.6.2. The relevant balance metrics are the COP velocity, RMS amplitude and RMS velocity, decomposed into their medio-lateral and anterior-posterior components.
To compute a balance score between 0 and 100 for each of these features, a frame of reference for the maximum achievable score must be defined. In a normal distribution, probability density is highest around its mean, declining symmetrically in both directions, hence proximity to the distribution's mean must correlate to higher scores. Given the probability density function of a univariate ND, shown in equation 4.2, when calculating the probability density for the mean μ, it is reduced to the expression shown in equation 4.5.

$$PD_{max} = \frac{1}{\sigma\sqrt{2\pi}} \tag{4.5}$$

Maximum probability density for a univariate ND

The probability density function for a univariate ND, as presented in equation 4.2, is utilized to compute the probability density for a new observation x. Finally, equation

4.6 shows how to achieve a score between 0 and 100 for a new observation x, where PD_{max} refers to the maximum value for probability density of the distribution and *PD(x)* represents the probability density of the new observation.

$$BS(x) = \frac{PD(x)}{PD_{max}} * 100 \tag{4.6}$$

Individual balance metric scoring

In a similar fashion, equation 4.7 shows how the deviation from the mean of a given distribution is computed. If this deviation is positive, the new value over-steps the mean of the underlying distribution, hence if the new value is negative, the mean is under-stepped.

$$BS_{diff}(x) = \begin{cases} \frac{PD_{max} - PD(x)}{PD_{max}} * 100 & \text{if } PD(x) - PD_{max} < 0 \\ \frac{PD_{max} - PD(x)}{PD_{max}} * -100 & \text{otherwise} \end{cases} \tag{4.7}$$

Deviation of new observation from maximum score

This detailed information is only shown in the session feedback. Analyzing each parameter individually for each level would result in considerable evaluation effort. Furthermore, incorporating only COP data based on a low number tiles lacks expressiveness. Thus, this information in provided a summarized manner after the session.

Overall balance score

In addition to the individual balance metrics, allowing analysis of each feature individually, an overall balance score, incorporating correlations between the features, is calculated. This overall balance score attempts to quantify balance ability when performing directional weight-shifting exercises, resulting in a score from 0 to 100. The distribution's centroid, or mean, is associated with the highest possible score of 100, decreasing steadily along the distribution. Hence, the less a new observation resembles the underlying distribution, thus healthy balance ability, the lower the score must be.
Therefore, the Mahalanobis distance, described in section 4.6.2, is utilized as distance measure that describes the distance between a new observation and the underlying MVND. The Mahalanobis distance for each level, only considering left-sided weight shifts, is visualized, sorted decreasingly in figure 4.46.

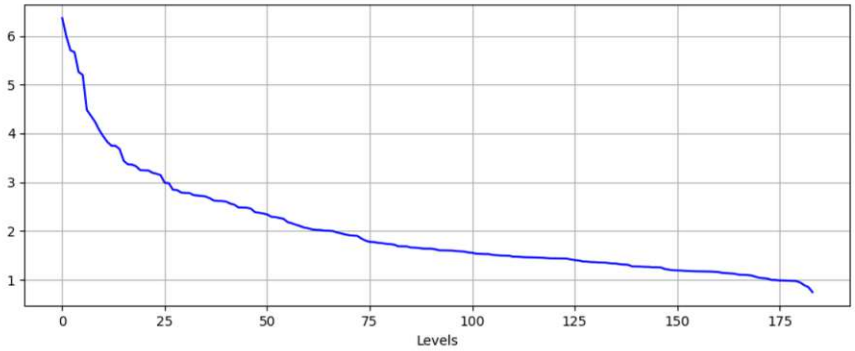


Figure 4.46: Mahalanobis distance for all left-sided weight shifts in training data

The further a new observation deviates from the distribution, the lower the overall balance score, thus the Mahalanobis distance is inversely proportionate to the achieved score. Hence, to translate the Mahalanobis distance into a scoring function, high distance values have to correlate with low scores. The simplest way to achieve this is to invert the distances for each level, shown in figure 4.47.

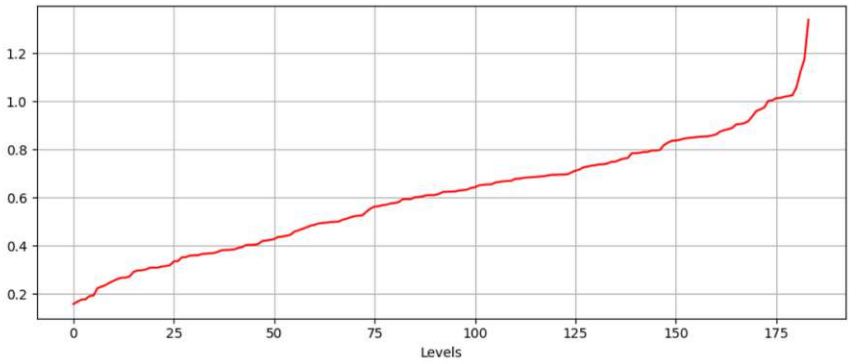


Figure 4.47: Inverted Mahalanobis distance for all left-sided weight shifts in training data

Inversion achieves the desired effect of associating low distances from the model with high scores. Although normalizing the values of inverted Mahalanobis distances into a range of 0 to 100 yields scores that fit the underlying distribution, the decline in score is considered too steep, even for healthy individuals. Taking into account the prototype is designed for players with impaired balance ability, the degradation in score must not be too steep, because this would result in players with low balance ability to always achieve a score of zero, rendering the balance score computation and the subsequent level generation unfeasible.
Thus, a lower threshold for the Mahalanobis distance has to be defined, acting as a cut-off point for score computation, normalizing scores in a broader interval while maintaining the underlying distribution. Any Mahalanobis distance above this threshold receives

a non-zero score, whereas distances below it convert to zero points. This threshold must yield high scores for individuals with healthy balance ability while simultaneously providing a sufficient margin for improper exercise execution considering players with balance impairments. This lower scoring threshold is defined as stated in equation 4.8, where μ_{Mah} constitutes the mean and σ_{Mah} the standard deviation Mahalanobis distances to the underlying model across all training samples.

$$LST = \mu_{Mah} + 5 * \sigma_{Mah} \tag{4.8}$$

Lower scoring threshold for overall balance score

The proposed threshold was empirically validated by performing test runs with healthy individuals, attempting to achieve lower scores by purposefully flawed execution of weight shifts, e.g. by performing jerky movements, shifting weight into the wrong direction or intentionally moving unusually slow. Equation 4.9 shows how the LST is incorporated into the final overall balance score computation, where *Mah(x)* refers to the Mahalanobis distance of a new observation x.

$$BS = \max(0, 1 - \frac{Mah(x)}{LST}) * 100 \tag{4.9}$$

Overall balance score

Figure 4.48 visualizes the overall balance scores for all left-sided weight shifts within the training data. It becomes apparent, that function more closely aligns with the Mahalanobis distances for left-sided weight shifts, shown in figure 4.46, but scaled into the appropriate interval and inverted accordingly.

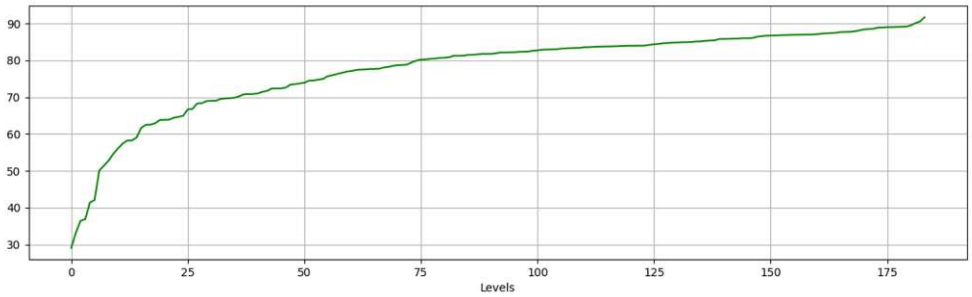


Figure 4.48: Overall balance score for all left-sided weight shifts in training data

CHAPTER 5

Evaluation phase

After designing and developing the prototype, an evaluation phase was conducted. During this phase, *Walk in the park* was integrated into four balance rehabilitation sessions of older adults with balance impairments. For each session, a physiotherapist was present to ensure patient safety, correct execution of the performed weight shifting exercise and to provide professional feedback regarding the design of game mechanics, as well as balance scoring and level generation.
Before each session, the patients were informed that no personal data was collected and that the data gathered during the session is anonymized. Furthermore, they received an introduction to the prototype and the correct execution of the weight shifting exercise. Each patient played the game for as long as they felt comfortable, which results in an uneven number of level data collected for each session. This does not affect the evaluation process in any way, although it limits comparability of results between the patients, because some only have performed few weight shifts in their strongest direction. After each session, patients were handed a questionnaire to gather feedback for evaluating functional and non-functional requirements regarding game design and mechanics. The results of these questionnaires can be found in section 5.3. This section also describes the questionnaire results regarding game design evaluation, which was conducted with the physiotherapists. Physiotherapists additionally received a questionnaire focusing on performance of balance scoring and level generation, which is discussed in section 5.2. The evaluation was carried out once per patient, which limits the expressiveness of the results regarding the prototype's capabilities of measuring performance progress.
In the following sections, the individual balance rehabilitation sessions utilizing *Walk in the park* are described, followed by a discussion about the questionnaire results in subsequent sections.

5.1 Game sessions

Walk in the park was incorporated into four individual rehabilitation sessions with patients suffering from balance impairments. The physiotherapists that volunteered for helping design the prototype, Hanna Schlosser and Raphaela Borg, were supporting the final evaluation. The sessions with Hanna Schlosser took place at SMO Dornbirn, where *Walk in the park* was firstly played by patients with balance impairments, *Player 1* and *Player 2*. Another two sessions were conducted with Raphaela Borg at the balance rehabilitation facilities of aks Vorarlberg, where the prototype was played by *Player 3* and *Player 4*. Figure 5.1 shows *Player 3* and *Player 4* play *Walk in the park* during their balance rehabilitation session. The following sections firstly describe a *Walk in the park* session with Raphaela Borg to highlight how the prototype evaluates healthy balance ability. The subsequent sections provide insight into the individual sessions for each impaired player and discuss prototype behaviour in more detail.

(a) Patient in post-stroke rehabilitation (*Player 3*)

(b) Patient suffering from general balance impairment (*Player 4*)

Figure 5.1: Evaluation session with patients at aks Vorarlberg

5.1.1 Evaluation of healthy balance ability

To reason about balance ability of impaired patients, it is necessary to firstly discuss how the prototype evaluates healthy balance ability, hence which scoring range is associated with properly executed weight shifts and how individual balance metrics are utilized to gain further insight. Each of the following sections, describing the individual sessions with impaired players, is accompanied with two plots, a statokinesiogram encompassing

raw COP data of all completed levels and an overall balance score progression. The following paragraphs describe how these visualizations are utilized for evaluation purposes. Furthermore, an introduction to the expressiveness of overall balance score and individual balance metrics is provided.

Statokinesiogram Figure 5.2 shows the statokinesiogram of a *Walk in the park* session executed by Raphaela Borg to exemplify a sway path for a healthy individual. The statokinesiogram shows a projection of the COP measurements onto a 2D pane. Directional weight shifts are color-coded, where blue corresponds to left, red to right, yellow to backwards and green to forwards weight-shifts. The gray lines visualize weight shifts on connecting tiles, which are not evaluated. The COP velocity-based features, used to model healthy balance ability in this thesis, can not be depicted using a statokinesiogram. Even conclusions regarding the COP RMS amplitude, described in section 2.6.2, are difficult to draw based on the overall sway path. Although expressiveness of the visualized sway path regarding performance evaluation is limited, it represents the raw weight shift data, which provides insight into the player's overall movement pattern and stimulates discussion about prototype enhancements.

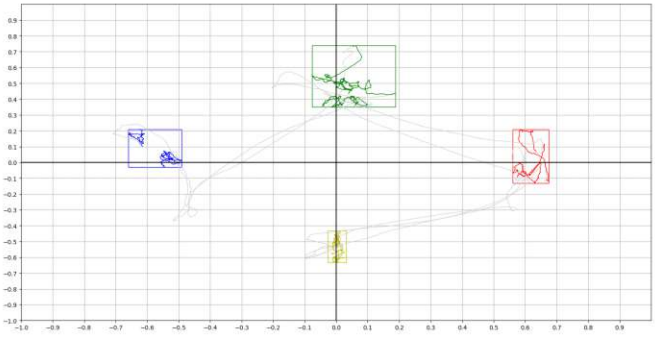


Figure 5.2: Statokinesiogram containing all levels performed by Raphaela

When comparing Raphaela's statokinesiogram with those of balance impaired patients in the following sections, differences in sway area, indicated by the color-coded rectangle, and the mean COP, referring to the mean displacement of COP measurements from the center, become apparent. As discussed in section 2.6.2, a small sway area is associated with increased ability to maintain balance, whereas large sway areas indicate a decreased ability to sustain an upright stance. Although the model for overall balance score computation does not incorporate the sway area as a defining feature, the sway area can be utilized as an additional metric to support conclusions based on the achieved score. Furthermore, comparing the distance of sway paths/areas to the center shows that the mean COP amplitude for Raphaela is higher for all directions, which indicates more intense weight displacement during exercises. Based on Raphaela's achieved overall balance score, described in more detail in the following, more intensive weight shifts correlate to a higher degree with the underlying model, thus yield higher scores.

Score progression During Raphaela's *Walk in the park* session, four levels were completed with a total of two tiles for left-sided weight shifts and three tiles for every other direction, resulting in 11 tiles total. For players without balance impairments, correctly performing the weight shift exercises, scores of 80 and above for each direction should be consistently reachable. Figure 5.3 shows the score progression of Raphaela's session. The horizontal axis indicates the completed levels. A circle signifies that the level contained at least one tile assigned to the respective direction. The worst score was obtained on the first level for a forwards weight shift, resulting 76.2 points. For the remainder of levels, Raphaela achieved scores of 80 and above for every single tile.

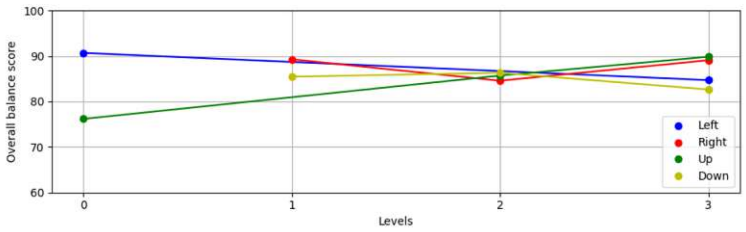


Figure 5.3: Score progression for each completed level for Raphaela

Overall balance score At the end of a session, an overall evaluation is computed, averaging the achieved scores over all completed levels. For Raphaela, this resulted in a score of 87.7 for left, 87.7 for right, 83.9 for forwards and 84.8 for backwards weight shifts. Even though Raphaela is a professional physiotherapist and performed the weight shifts exercises in excellent fashion, she did not receive more than 91 points for any level or direction. There is a multitude of reasons for this. Firstly, the prototype's overall balance score computation is not designed to estimate or compare healthy individuals' balance ability, but rather to quantify the divergence of impaired patients from a healthy baseline. Furthermore, this baseline was constructed from different individuals with varying physical constitution. Hence, a weak threshold has to be defined when interpreting the overall balance score to encompass data fuzziness. This threshold is considered weak, since even healthy individuals, when performing subpar weight shifts, can score below it for individual levels, but are not expected to do so repeatedly. Consecutively achieving scores below the given threshold indicate an inability to perform weight shifts in the associated direction. Based on the underlying training data and Raphaela's session, a score of 80 points and above should be achievable for healthy individuals.

Individual balance metrics For Raphaela, analysis of individual balance metrics did not present obvious weaknesses. In case of impaired patients that are discussed in the following, looking into balance metrics becomes more relevant. Each directional weight shift is associated with three balance metrics - COP velocity, COP RMS velocity and COP RMS amplitude - decomposed into their medio-lateral and anterior-posterior

components. These metrics provide vital insight into a player's balance performance. The underlying model expects the values for each of these metrics to be within a certain range. Deviating heavily from this range results in poor performance scores. A lower COP velocity is associated with controlled weight displacement, whereas high COP velocities suggest the opposite. An exceptionally low COP velocity indicates missing confidence or fear of performing weight shifts into the specific direction, whereas a particularly high COP velocity suggests a loss of control over one's weight displacement. When considering COP velocity as swiftness of weight shifting, COP RMS velocity can be translated as acceleration, or a measure of variation in velocity, during weight shifts. A high variability in COP velocity, hence increased COP RMS velocity, suggests jerky weight shifts, whereas low values correlate with steady and controlled weight displacement. Consistently high COP RMS velocity suggests hasty and uncontrolled weight displacement. The COP RMS amplitude constitutes a measure of stability, where lower values are associated with an increased ability for postural control, thus non-shaky, directional weight shifts. These individual balance metrics are discussed in greater detail in section 2.6.2.

The following sections describe evaluation sessions performed with impaired patients, outlining findings during their sessions, reflecting on the prototype and analyzing if the achieved scores adequately reflect their impairments.

5.1.2 Player 1

The first balance impaired person that played *Walk in the park* was a recovering stroke patient. The player was suffering severe balance impairment and was not able to maintain an upright stance without external support. They were able to perform the weight shift exercises while holding on to a chair in front of them. *Player 1* completed 7 levels consisting of 46 tiles. Figure 5.4 shows the score progression during the session. They progressed very promptly and finally achieved balance scores that were higher than expected.

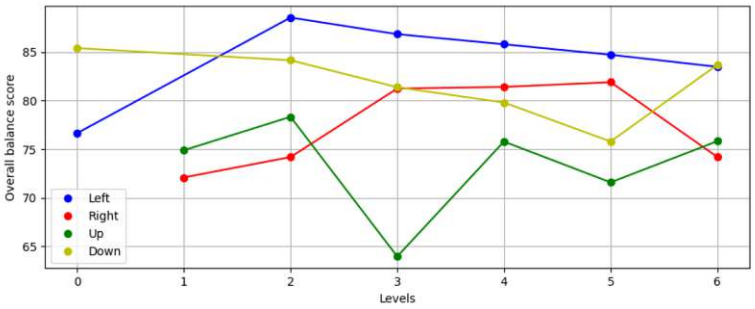


Figure 5.4: Score progression for each completed level for *Player 1*

Due to the assistance, provided by holding on to a chair, the player was capable of executing weight shifts in a controlled manner, weakening expressiveness of the resulting

overall balance scores. *Walk in the park* was never before tested on players utilizing support structures. With the provided physical assistance, *Player 1* was able to perform the exercises in a way that came close to what the underlying model expected of a healthy person. An analysis of the individual balance parameters shows that 23 out of 24 metrics were under-stepped, providing further evidence that the player shifted their weight in a very controlled fashion. Even though most metrics were below the mean, deviations were not drastic. The highest deviation from the model was found in COP velocity, which correlates with the swiftness of weight displacement. The comparatively small sway areas in *Player 1*'s statokinesiogram, shown in figure 5.5, reinforce the above conclusions.

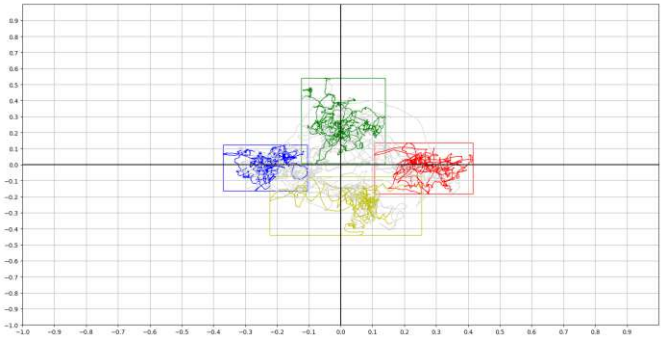


Figure 5.5: Statokinesiogram containing all levels performed by *Player 1*

Overall, *Player 1* achieved balance scores of 84.4 for left, 77.5 for right, 73.4 for forward and 81.7 for backwards weight shifts. All of these values are close enough to the weak threshold, discussed in section 5.1.1, that overall balance ability could be considered healthy. During the session, *Player 1* completed 9 tiles for left-sided and backwards weight shifts respectively and 14 tiles for each right-sided and forwards weight shifts, resulting in a total of 46 tiles across 7 levels. The correlation between achieved scores and tile configuration of the generated levels provides evidence that the prototype fulfills requirement **LevGen-1**, defined in section 4.1.1, generating levels that encourage weight shifts in the player's weakest directions. The player achieved a slightly better score for left-sided weight shifts, which is in-line with patients balance impairments, although no conclusions should be drawn due to the minor difference. This session provided evidence that *Walk in the park* does not provide meaningful insight into balance performance when players are supported by external aids. Nonetheless, the data suggests that, even when incorporating aid for postural stability, weaknesses in weight displacement may be identified.

5.1.3 Player 2

Player 2 was also a post-stroke patient, suffering from general balance impairment, undergoing physical therapy for balance rehabilitation. They completed 10 levels, consisting of 32 directional tiles in total, which is less tiles than *Player 1*, even though they played

more levels. Level size increases with players' experience gains, which are correlated with score progress, as discussed in more detail in section 4.4.5. *Player 2*'s score progression is visualized in figure 5.6. Due to consistently achieving zero points for forwards weight shifts throughout level #0 to #4, *Player 2*'s experience gains were minor, thus level generation provided the smallest possible levels for them. After five levels of receiving no points for their forward weight shifts, *Player 2* managed to improve their performance after some encouragement and reassurance. Another relevant detail is that during the *Walk in the park* sessions, players with balance impairments were supported from the back, resulting in them having more confidence in their back-sided weight shifts due to the reduced risk of falling. Apart from *Player 1*, none of the players utilized additional support to stability during their weight shifts, which leads to a more realistic balance assessment.

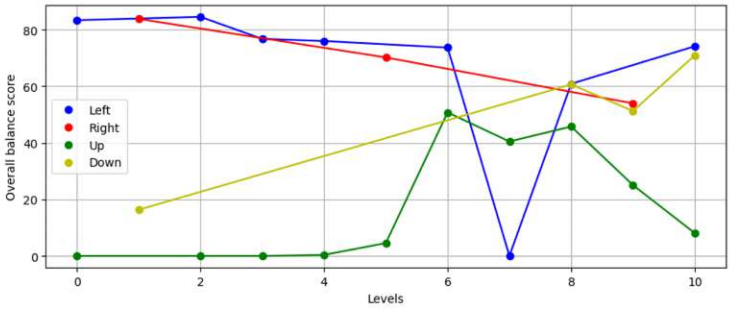


Figure 5.6: Score progression for each completed level for *Player 2*

Player 2 achieved comparable overall balance scores for their left- and right-sided weight shifts of 69.4 and 69.3 points respectively, indicating no weaker side for medio-lateral weight shifts. Comparatively large sway areas for forwards and backwards weight shifts in the statokinesiogram, shown in figure 5.7, indicate a weakness in anterior-posterior weight shifts, which is validated by the performance outcomes. *Player 2* received an overall balance score of 22.8 for forwards and 38.6 points for backwards weight shifts. They completed 7 tiles for left-sided, 3 right-sided, 14 forwards and 8 backwards weight shifts, further reinforcing confidence in the level generation logic, especially regarding requirement **LevGen-1**.

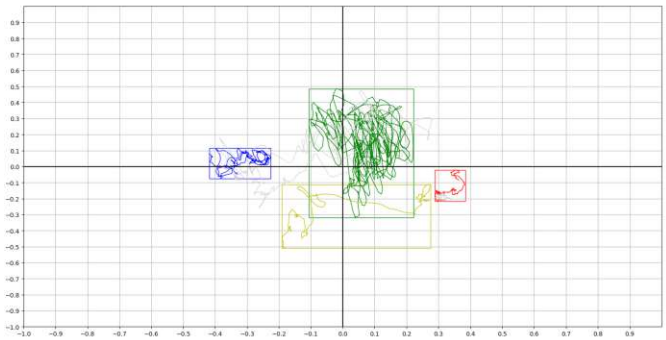


Figure 5.7: Statokinesiogram containing all levels performed by *Player 2*

Analysis of individual balance metrics suggests that especially the anterior-posterior components during forwards and backwards weight shifts deviated heavily from the model. Throughout level #0 to #4 *Player 2* displaced their weight in short bursts forwards in an attempt to push the in-game avatar upwards, where a controlled and steady weight displacement is expected. Consequentially, the anterior-posterior components of COP velocity, COP RMS velocity and COP RMS amplitude deviated heavily from the model data. In contrast to *Player 1*, *Player 2* over-stepped each of the balance parameters due to their uncontrolled, jerky forwards weight shifts within the first five levels. Thus, the prototype is capable of detecting these irregular bursts in weight displacement and adequately penalize scoring.

5.1.4 Player 3

The third evaluation session was conducted with *Player 3*, who was also recovering from a stroke and suffering from general balance impairment. Figure 5.8 shows the performance progression throughout the session. They completed 9 levels, consisting of 60 directional tiles in total, achieving scores of 56.8 for left, 61.2 for right, 45.7 for forwards and 56.1 for backwards weight shifts. None of these scores are close to the threshold for healthy balance ability, thus the scores adequately reflect the player's general balance impairment. Based on player performance, they completed 3 tiles for left-sided weight shifts, 21 tiles for-right sided shifts, as well as 17 and 19 tiles for forwards and backwards weight shifts respectively. Due to a score of 89.4 in the first level for their left-sided weight shift, *Player 3* was only presented with a low number of tiles for this direction. Thus, the score for left-sided weight shifts is only based on 3 tiles, rendering it less meaningful than their other scores. This could be considered a weakness of the prototype, although with larger levels, stronger directions for the player are re-introduced, as can be seen in level #8 and #9 in figure 5.8.

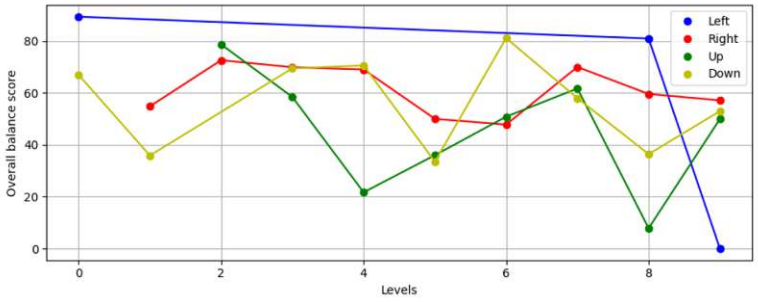


Figure 5.8: Score progression for each completed level for *Player 3*

The player's statokinesiogram, shown in figure 5.9, accentuates a general balance impairment, represented by sway areas larger than those of *Player 1* or *Player 2*.

Figure 5.9: Statokinesiogram containing all levels performed by *Player 3*

An analysis of individual balance metrics shows a general trend of over-stepping each feature, suggesting a weakness in control during weight shifts. For both left- and right-sided exercise execution, *Player 3* continuously over-stepped especially the medio-lateral components, indicating uncontrolled and jerky movement, although no compensation by displacing weight in anterior-posterior directions. The general trend that forwards and backwards weight shifts yield lower scores is also prevalent for *Player 3*. Especially weak performances throughout the sessions for forwards weight shifts become apparent. Weight displacement in forwards direction might have been considered more risky since there was no additional support.

5.1.5 Player 4

The last evaluation session was carried out with *Player 4*, also suffering from general balance impairment without disclosing the underlying pathology. The general impairment is reflected by their overall results of 56.3 points for left-sided, 60.8 for right-sided, 33.2 for forwards and 66.5 for backwards weight shifts. Level generation provided only four

tiles for left-sided weight shifts due to a high score in the first level, similar to *Player 3*. Considering the other directions, level generation provided 16 tiles for right, 29 for forwards and 11 for backwards weight shifts. Figure 5.10 visualizes the score progression for *Player 4* throughout their session.

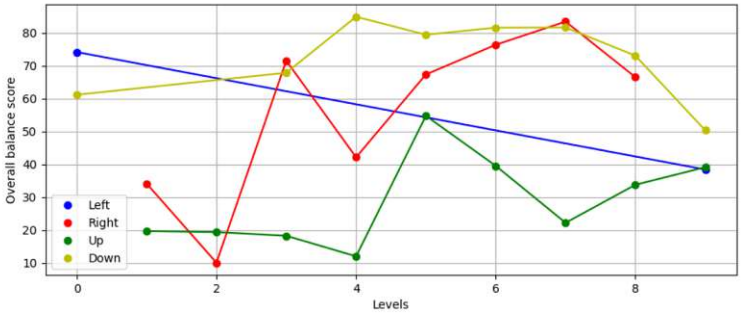


Figure 5.10: Score progression for each completed level for *Player 4*

Due to the low number of left-sided weight shifts performed during the session, data for comparing left- and right-sided weight shift performance is insufficient. Similarly to other balance-impaired patients, forwards weight displacement is evidently hardest to perform. The overall score distribution underlines the player's overall balance impairment. Especially expressive is the sway area visualized in the player's statokinesiogram in figure 5.11.

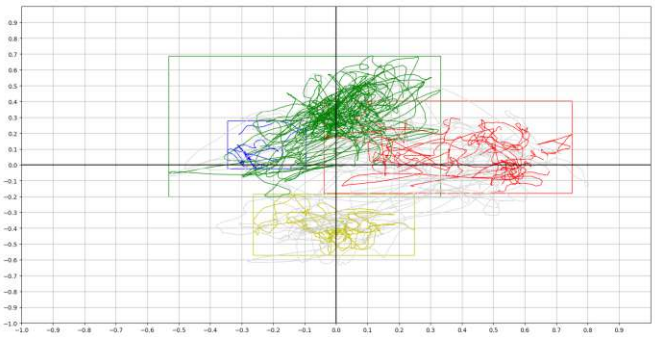


Figure 5.11: Statokinesiogram containing all levels performed by *Player 4*

All 24 individual balance metrics were consistently over-stepped during each level performed by *Player 4*, which suggests significant deficits in overall balance control. When comparing left-sided and right-sided weight shifts, it shows that for left-sided weight shifts especially medio-lateral balance components were significantly over-stepped, whereas anterior-posterior components were closer to the expected values, which suggests a lesser degree of compensation through weight displacement forwards and backwards. In comparison, right-sided weight shifts show a larger deviation from the mean in anterior-posterior

components, indicating directional weight shifts being accompanied by weight displacement forwards and backwards, indicating poor postural control.

The analysis of evaluation sessions with impaired patients provides insight into the *Walk in the park*'s ability to quantify weight shift quality and generate personalized levels based on the achieved scores. Understanding the prototypes capabilities acts as a basis for the evaluation of questionnaire results in the following sections.

5.2 Balance scoring and level generation evaluation

Performance of balance scoring and level generation was evaluated using a catalog of four acceptance criteria derived from a subset of the requirements gathered during the research phase, discussed in section 4.1.1. The statements, scores given by the physiotherapists and requirements the statements are derived from are presented in table 5.1. The original questionnaire in German language can be found in figure A2, located in the appendix. The relevant design criteria and requirements for this evaluation are **Lit-3** and **Lit-5**, validating whether balance scores reflect patient impairments (Lit-3) and responsiveness of level generation to those impairments (Lit-5). After the sessions, described in section 5.1, were concluded, the physiotherapists were asked to grade the prototypes performance in balance scoring and generating personalized levels for the patients on a scale from 1 (not satisfactory) to 5 (very satisfactory).

Code	Statement	Score	Req.
L1	The balance evaluation, generated by the application, reflects the patient's balance impairments.	4.0	Lit-3
L2	Level generation adapts based on the patient's balance impairments (prioritizing the weakest direction).	4.0	LevGen-1
L3	The application sufficiently adapts level difficulty (path length) based on patient's proficiency level.	4.0	Lit-5, LevGen-3
L4	The applications progressively leads the patient to their proficiency level.	5.0	Lit-5, LevGen-2 LevGen-4

Table 5.1: Evaluation results of balance scoring and personalized level generation

L1 This constitutes the most significant acceptance criteria, because **L2** is directly correlated with the balance evaluation, whereas criteria **L3** and **L4** are indirectly dependent on it. These relationships are discussed in the following paragraphs respectively. The prototype's balance evaluation achieved a score of 4.0, which means it's capabilities of evaluating the patient's balance impairments are sufficient. During the first session, *Player 1* achieved very high scores in all directions and their weaker side was just barely recognizable in the session evaluation result. This provided evidence that even with

severe balance impairments the prototype might incorrectly evaluate the patient's balance ability as healthy, when the weight shifting exercise is performed with external support. The patient's total scores in all directions are in the range of healthy individuals, yet a slight but insignificant weakness in their weaker direction was found. This session showed that *Walk in the park* does not provide an accurate overall balance estimation for players that can not perform unassisted directional weight shift exercises. For the remainder of sessions, balance scoring satisfyingly captured weaknesses in balance ability. The sessions with *Player 3* and *Player 4* showed that a general balance impairment could be captured within a single session. For *Player 2*, a weakness in balance ability in anterior-posterior direction was identified. The insight gathered during the sessions, described in section 5.1, were in line with balance estimations from the physiotherapists. An issue to address is the high variability in balance score throughout the session. This can be explained by the lack of experience interacting with the WBB or performing the exercises for the first time in this setting. Assuming patients become accustomed to interacting with the WBB and gain more experience performing the weight shifting exercise, this would eventually lead to reduced fear and more ambitious weight shifts, resulting in more appropriate estimation of balance ability.

L2 The second acceptance criteria for level generation is concerned with adapting the level design, prioritizing the patient's weakest direction, thus force the patient to shift their weight towards their most impaired direction. In other words, the generated path of a level should be longest in the patient's weakest direction. Acceptance criteria **L1** is crucial in achieving this behavior, because the path generation heuristic, described in section 4.4.5, directly translates balance evaluation results into the number of path tiles in the respective direction. A score of 4.0 suggests that this acceptance criteria is fulfilled in a satisfying manner, but there is still room for improvement. If a patient has severe problems shifting their weight in a certain direction, the level generation engine will provide levels forcing the patient in this specific direction. During the evaluation three out of four patient showed problems shifting their weight to the front, forcing the patient's toes to bear weight. In fact, balance scores for weight shifts in anterior-posterior direction were lower relative to the scores in medio-lateral direction for all patients. This lead to levels primarily forcing the patients to weight-shift in anterior-posterior directions. This behavior is intended, yet could be improved by randomizing to a certain degree every few levels to increase diversification of levels. *Player 2* negatively commented on their levels being too focused on their weakest direction, which could be demotivating. Due to the fact that most patients only played for ten minutes before being exhausted, they did not reach a level of experience that would have generated larger levels. An increasing level size directly leads to a more heterogeneous level configuration in terms of directional changes. Hence, the problem of over-fitting levels that ultimately result in demotivation and boredom can be circumvented by gaining experience during more extended play.

L3 As level layout is dependent on balance scores for each direction, the level size, and subsequently path length, is dependent on player experience. Level difficulty in *Walk in the path* is solely based on length of the generated path. The discussion of **L1** mentions that **L3** is indirectly dependent on the balance evaluation. The reasoning behind this is that the amount of experience points gained or lost per level increases or decreases relative to the total performance for the completed level. Thus, the more a patient improves by playing, the more experience points they will obtain. The score of 4.0 for this acceptance criterion indicates that difficulty adjustment based on player experience is working satisfyingly. Unfortunately during the first evaluation session, *Player 2* struggled with achieving a non-zero score for forwards weight shifts, resulting in minimally sized levels, although the patient already surpassing experience required for larger levels. For the other sessions, level difficulty scaled sufficiently with player experience. The concrete scores for each session for balance impaired players and associated tile configurations provided by *Walk in the park* are discussed in more detail in the individual game session descriptions in section 5.1.

L4 Progressing in the game is directly related to experience points gained or lost during gameplay. Every session starts by generating a level incorporating all previous session results. During a session, scores for levels finished in the current session are incorporated to provide historic and current data into the level generation process. This acceptance criteria was deemed very satisfactory by both physiotherapists with a score of 5.0.

Design criterion **Lit-1** was not part of the evaluation sessions, because none of the impaired patients had severe one-sided balance impairments, thus there was no need for input signal adaptation via calibration. As shown in the final screen flow in section 4.4.4, every new session starts with calibration. Even though it was not needed during the sessions, calibration was verified to work as intended. The concrete implementation details are discussed in section 4.4.6.
Design criteria **Lit-2** and **Lit-4** are not incorporated into the above statements, because these requirements can only be evaluated directly by the patients. After each session, players received a questionnaire to evaluate *Walk in the park* specifically regarding design criteria **Lit-2** and **Lit-4**, discussed in section 4.1.1, and a patient-specific subset of requirements, listed in the final requirements catalog in section 4.1.1. Hanna Schlosser and Raphaela Borg also evaluated the SG aspects of *Walk in the park* in a separate questionnaire. The results of these questionnaires are discussed in the subsequent sections.

5.3 Gameplay evaluation

In the following, table 5.2 presents evaluation results from the patient's perspective. Accuracy of the provided statements was graded on a scale from 1 to 5, where 1 indicates unsatisfactory implementation and 5 corresponds to a very satisfactory incorporation of the associated requirements into the prototype. The original questionnaire in German language can be found in the appendix, see figure A3. A printed out version of this questionnaire was handed to the patients immediately after completing their session. Every participant was able to independently complete the questionnaire.

Code	Statement	Score	Req.
P1	The game is not too exhausting and properly adapts in level difficulty.	4.5	Lit-4, Lit-5
P2	The weight shifting exercise performed in the game is not too easy.	4.5	Lit-4
P3	I would like to play the game more often in my therapy sessions.	4.5	Lit-2
P4	I was motivated to play some more.	3.75	Lit-2, Lit-5
P5	The game does not frustrate me.	5.0	Lit-2, Lit-4
P6	I would feel comfortable playing the game at home.	4.25	Lit-2, Lit-4
P7	It is easy to follow the gameplay mechanics.	5.0	Lit-4
P8	The feedback after each level and at the end of the session helped me better understand my deficiencies in balance ability.	4.5	Lit-3, Int-5
P9	The evaluation of my balance ability was fitting.	4.75	Lit-3
P10	The game was tiring but doable.	4.75	Lit-5
P11	I was slowly brought towards my proficiency level.	4.0	Lit-5, LevGen-4

Table 5.2: Evaluation results of gameplay from the patient's point of view

Based on the questionnaire results, *Walk in the park*'s in-game mechanics and overall game design are considered simple and easy to follow for patients in post-stroke rehabilitation (**P1**, **P2**, **P5**, **P6**, **P7**). The weight shifting exercise was considered challenging, yet doable by the impaired players (**P1**, **P2**). The performance feedback after each level was deemed helpful and vital in understanding their own balance deficiencies (**P8**, **P9**). Progression throughout the game felt natural to the players, except for *Player 2*, who initially struggled with performing forwards weight shifts (**P11**). Duration of evaluation sessions was approximately 10 minutes on average, after which players expressed that they felt exhausted, but not frustrated (**P4**, **P10**). Results regarding motivational aspects of the game are promising. The majority of players wanted to play the game more often in future therapy sessions **(P3)**, stressed that gameplay was not frustrating (**P5**) and they would even feel comfortable playing *Walk in the park* at home (**P6**).

Table 5.3 contains questionnaire results regarding design criteria and requirements for SG aspects of *Walk in the park*. This questionnaire is aimed to cover topics relevant for the physiotherapist. The original questionnaire is located in the appendix, see figure A2. The scoring metrics are equivalent to the previous questionnaires.

Code	Statement	Score
S1	The game mechanics are designed in a simple fashion and the patients are not overwhelmed.	5.0
S2	The patient's progress regarding performance during the session is well indicated.	4.5
S3	There is no recognizable delay between in-game movement and weight shifts on the balance board.	5.0
S4	The weight shifting exercise performed for balance evaluation with the application is an effective exercise for balance rehabilitation.	4.0
S5	I could see myself including the application as an addition to regular balance rehabilitation measures.	4.5

Table 5.3: Evaluation results of gameplay from the physiotherapist's point of view

S1 This statement aims to evaluate design criterion **Lit-4** from the physiotherapist's point of view. They thoroughly observed the patients during their sessions and concluded that *Walk in the park*'s gameplay mechanics are easy to understand even for elderly patients with impairments.

S2 *Walk in the park* provides performance feedback after every completed level, which was appreciated by the physiotherapists, thus fulfills requirement **Int-5**.

S3 Requirement **Int-8** is vital for the prototypes integrity, thus it was verified early on during prototype development. Nonetheless, this question aims further solidify confirmation of low input delay by the physiotherapists.

S4 Since *Walk in the park* only supports the execution and evaluation of directional weight shifts, utility of the application highly depends on effectiveness of the performed exercise. Physiotherapists deemed weight shifting exercises a vital part of balance rehabilitation, although not suitable for some patients. *Player 1*, for example, could only perform the weight shift exercises with additional support. In this case, the patient benefits more from performing other exercises.

S5 This question aimed at overall satisfaction with the prototype, rather than any specific design criterion or requirement. Overall, the response from both physiotherapists was very positive. The only concern was setup time of the application, because incorporating *Walk in the park* into balance rehabilitation sessions would require a computer

setup that allows for exercise execution in front of a monitor and a constantly running PC with a coupled WBB in order to keep setup time between patients to a minimum.

The evaluation phase overall shows that *Walk in the park* provides a motivating and engaging experience for its players. Furthermore, its capability of estimating balance ability has been deemed sufficiently accurate to derive personalized game levels, tailored to the balance impairments of each individual patient. The following chapter reiterates on the contributions of this work and how it builds upon the current state of the art. Furthermore, it provides an in-depth discussion on the results of this thesis and accentuates some of the weaknesses exhibited by the proposed approach.

CHAPTER 6

Discussion

Walk in the park incorporates several aspects of WBB-based balance rehabilitation research and combines these into a new motivating and engaging SG, adjusting level layout and difficulty to the player's weaknesses in a novel fashion.
To achieve this, the resulting prototype was developed with the following three research questions in mind:

1. **What are the requirements for a SG in the context of balance rehabilitation?**
2. **What are the requirements for personalized level generation in balance rehabilitation?**
3. **Can machine learning techniques be utilized to derive personalized game levels from patient movement patterns extracted from a WBB?**

The following sections provide further discussion of the results for each of the research questions. Finally, weaknesses of the prototype are discussed in section 6.4.

6.1 Requirements for SGs in balance rehabilitation

To answer the first research question, two independent interviews were conducted with physiotherapists, specialized in balance rehabilitation. The first part of the interview process aimed to gain insight into the clinical practice of balance rehabilitation sessions to cooperatively define requirements for the prototype to be usable within such sessions. After concluding the interviews, a catalog of all requirements was compiled, which can be found in section 4.1.1. Subsequently, to gain further understanding of the relative importance of each requirements towards improving clinical feasibility of the prototype,

this catalog was sent to the physiotherapists for review. The physiotherapists were asked to weigh importance of each requirement on a scale from 1, meaning less important, to 5, which means the requirement is vital to prototype design.
In addition to the interviews, extensive literature research was conducted, which resulted in a catalog of five design criteria for SGs in balance rehabilitation. A more in-depth explanation of this phase, along with the resulting catalog, can be found in section 4.1.1.
In a subsequent discussion, the catalog of weighted requirements from the interviews and the list of design criteria from the literature research phase, were evaluated, resulting in the final requirements catalog for SGs in balance rehabilitation. Considerations for each decision were discussed in section 4.1.1.
To conclude, literature research resulted in sophisticated design criteria, used extensively throughout related work, describing vital aspects a SGs for balance rehabilitation must implement. In addition to the design criteria, the final, condensed catalog from the interviews describes a set of requirements that establish utility of the prototype in a balance rehabilitation session. *Walk in the park* was developed with these requirements catalogs at the core of each design decision, thus building on top of current state-of-the-art work in SGs in balance rehabilitation.

6.2 Requirements for personalized level generation in balance rehabilitation

The later part of the interview process aimed at answering the second research question regarding requirements for personalized level generation in balance rehabilitation. The process and the resulting requirements catalog is described in section 4.1.1.
LevGen-1 aims to counteract a common issue of patients with balance impairments. When struggling to perform weight displacement into a specific direction, patients build up anxiety, thus avoid doing so in day-to-day activities, which creates an over reliance on their stronger side. Based on the evaluation results, *Walk in the park*'s level generation mechanism sufficiently counteracts this compensation technique by creating levels that specifically target the patient's weak direction.
Incorporating historic session data allows creating levels that reflect previous performance and establish a baseline for the patient. Combining this data with intra-session performance, as described in requirement **LevGen-2**, allows creating personalized levels that match the patient's balance impairments.
Steadily increasing level difficulty, as described in requirement **LevGen-3**, is achieved by naturally progressing in experience score, creating larger levels with more tiles, and changes in overall balance estimation, resulting in varying level configurations.
Similarly to **LevGen-3**, slowly approaching player proficiency level during a session, as described in requirement **LevGen-4**, is achieved naturally by progressing in experience score and variations in overall balance score estimation.
Requirements for personalized level generation in balance rehabilitation are unique to this thesis. In related work, adaptation mechanisms to game design based on in-game performance are discussed. In the current state of the art, difficulty is modified by

speeding up gameplay and spawning more in-game objectives in harder-to-reach locations. None of the current state-of-the-art contributions provide adaptations to level layout based on objective performance scores. Thus, the catalog for personalized level generation in balance rehabilitation can be utilized as baseline for future research.

6.3 Machine-learning-based personalized level generation based on WBB sensory data

This section describes how the novel approach for balance score computation based on analysis of WBB-based sensory data and subsequent personalized level generation, presented in this thesis, relate to current state-of-the-art work. Describing the resulting solution and its relation to the state of the art aims to answer the third research question. Section 6.3.1 discusses the relation of *Walk in the park*'s balance score computation to the current state of the art and emphasizes its advantages. Subsequently, section 6.3.2 details *Walk in the park*'s capabilities to adapt to patients' balance impairments with a focus on comparing the prototype's personalized level generation with current state-of-the-art difficulty adaptation mechanisms in SG for balance rehabilitation.

6.3.1 Balance score computation

In their work on *WeHab* [1], *Kennedy et al.* compute a score that reflects overall balance ability during weight shift exercises. They conclude that utilizing their approach, they found no meaningful correlation between the player's balance impairments and the resulting scores. A possible explanation for this is that *Kennedy et al.* did not differentiate between weight shift directions. In case a balance impairment severely affects weight displacement in one direction but the patient has no problems shifting their weight otherwise, this could misrepresent overall balance estimation. Furthermore, computing an overall balance score without differentiating between directions omits vital information about the patient's balance ability.
Evaluation results of *Walk in the park* provide evidence that the balance scoring computation, implemented in this thesis, is capable of correctly capturing general balance impairment and weaknesses in directional weight shifts. Since *Kennedy et al.* did not disclose implementation details of their approach for balance score computation, there is no basis for further discussion. Nonetheless, *Kennedy et al.* provided therapists with quantitative feedback by computing balance metrics from COP data. Concretely, they computed the COP mean velocity and COP RMS amplitude to gain insight into patient's balance ability. These metrics are also part of the model *Walk in the park*'s balance score computation is based on and are also utilized in related work, e.g. the work of *Mertes et al.* [3], in combination with machine learning techniques.
In more recent research, discussed in section 3.3.1, *Mertes et al.* [3] utilize WBB-based COP data to classify patients as either fallers or non-fallers. The features chosen to make this distinction are also indicative of overall balance ability and, in combination with machine learning techniques, yield promising results. Hence, balance score computation

for *Walk in the park* also makes use of this feature set, comprising of COP mean velocity, COP RMS velocity and COP RMS amplitude. The work of *Bacciu et al.* [9] further solidifies the notion of richness in COP data by successfully approximating a patient's BBS based on a 10 second measurement.
As opposed to the current state of the art in quantifying balance ability, which is predominantly based on supervised machine learning techniques, *Walk in the park* utilizes a combination of unsupervised learning and distance measures to quantify balance ability. This is achieved by building a model representing healthy balance ability, measuring the distance of a new observation from the model and converting this distance into a score. This constitutes a novel approach with the benefit that data can be collected from healthy individuals instead of impaired patients, which allows sampling training data from a larger audience, thus it becomes easier to generate sufficient training data.
Based on the evaluation results, balance scores computed by the novel approach used in *Walk in the park* show a significant correlation with the balance impairments exhibited by the patients. The adaptation mechanism in *Walk in the park*, incorporating patient impairment and skill into the level design, utilizes these balance scores as parameters to generate personalized levels. The following section aligns the approach used in this thesis for adaptation to player impairments with current state-of-the-art work.

6.3.2 Adaptation to player impairment

A vital aspect of SG in balance rehabilitation is adjusting game difficulty to a player's level of skill. To create an understanding of how *Walk in the park* improves on the current state-of-the-art approaches for adaptation mechanics, the evolution of adapting difficulty to patient impairment needs to be outlined.
Firstly, in their work on *SilverBalance* [65], discussed in section 3.2.2, *Gerling et al.* introduce a mechanism to progressively adapt game difficulty within a session based on time. At the end of each session, the only metric to measure balance ability is the overall time spent in the level. In case of *SilverBalance*, difficulty is highly dependent on randomized spawning locations of in-game objectives, thus player skill might not correlate with their best time. An argument can be made that over the course of multiple sessions, the factor of randomness is mitigated. Nonetheless, deliberately ramping up difficulty solely based on time might impact player motivation due to bad luck in the randomized objectives, or in case of severe balance impairment, consistently failing after a short amount of time.
Further contributions in *eBaViR* [7] and *IGER* [8], discussed in sections 3.2.3 and 3.4 respectively, introduce an enhancement of *SilverBalance*'s adaptation mechanism, incorporating an overall received score that reflects performance and accomplishments, such as the number of collected in-game items. This constitutes a more objective and direct representation of balance performance, yet it still lacks expressiveness, because patient skill is associated with solving in-game objectives instead of their ability to shift their weight in a stable and controlled manner. There might be a correlation between maximizing achievement of in-game objectives and balance ability, but this approach omits vital insight about stability and correctness of weight shifts that can be gained

from COP measurements. An optimal adaptation mechanism would quantify the player's balance ability based on their COP measurements during gameplay and adjust difficulty accordingly.
Walk in the park improves upon the state-of-the-art in adaptation mechanisms, because it quantifies balance ability based on COP data and adjusts level layout and difficulty based on player experience and detected weaknesses. This approach uses the raw data to reason about patient balance ability instead of utilizing abstract in-game objectives, which constitutes a more direct and objective approach. The resulting balance score is then used to parameterize a level generation heuristic, adjusting level layout to fit the player's impairments.

Evidently, *Walk in the park* overlaps with many areas in *WiiHabilitation*. Its main contribution to the current state of the art is the adaptation mechanism for personalized level generation. The usage of unsupervised learning techniques in combination with distance measures to compute a balance score for each individual directional weight shift is at its core. Providing a motivating and engaging SG that adapts difficulty by generating levels based on the player's actual quantified capability to perform directional weight shifts instead of abstract in-game objectives differentiates *Walk in the park* from the current state of the art in SGs in balance rehabilitation.

6.4 Weaknesses of the prototype

The evaluation results suggest that *Walk in the park* satisfyingly reflects balance weaknesses in patients. Reasoning about the presented data is vital in analyzing and understanding these weaknesses. Section 5.1.1 discusses a threshold that defines a cut-off point at which a balance score is considered healthy. Scoring below this threshold needs to be followed by interpretation of individual balance metrics to analyze the player's balance impairments in the affected direction. Section 5.1.1 and the subsequent evaluation sessions explain the individual balance metrics and exemplify analysis of balance weaknesses based on them. All necessary information for further analysis is displayed in the session overview screen, shown after completing a session. Although these evaluation results are rich in information, their presentation could be improved. During the evaluation, physiotherapists suggested that individual balance metrics could be named more closely after their intent instead of their concrete feature descriptor. For example an appropriate synonym for COP mean velocity could be swiftness and COP RMS velocity could therefore be called acceleration. Furthermore, clarifying the rationale behind over- and under-stepping of parameters in the game would have been appreciated by the physiotherapists. Thus, a weakness of *Walk in the park* is presentation of the overall session results, especially regarding individual balance parameters. Even though *Walk in the park* is designed to be utilized with trained professionals present, that have been introduced to the prototype beforehand, it would constitute an improvement if evaluation results were presented in a more comprehensible fashion.

Before each evaluation session, players were informed that they do not have to move towards coins or monsters that spawn along the generated path, because interaction with these in-game objects happens automatically when moving onto the same tile. In some cases, players still adjusted their weight shifts to steer towards those in-game objectives. These adjustments were not severe, but recognizable. To prevent this, instead of allowing free maneuvering of the in-game avatar within the path's bounds, movement could be locked in the path's center and restricted to only the direction assigned to the tile. The impact of this issue on the evaluation results is considered minor, but it could potentially further improve expressiveness of results.

For the development of this prototype, a tile-based level design is utilized, hence balance score computation is performed on a per-tile basis, then averaged over the amount of tiles for the given direction. This simplifies level generation but also leads to similar level configurations in early stages of the game. Converting balance scores into relative path lengths could yield a more appropriate reflection of player balance impairments.

Lastly, the number of participants and overall duration of evaluation can be considered a weakness, reducing expressiveness of the achieved results. Finding volunteering physiotherapists that specialize on balance rehabilitation turned out to be a difficult task. Fortunately, Hanna Schlosser and Raphaela Borg devoted some of their valuable time to help design, develop and evaluate *Walk in the park*. Their recruitment efforts finally resulted in a total of four balance impaired patients that wanted to participate in the evaluation sessions. During these sessions, patients were asked to play as long as they felt comfortable, which was approximately 10 minutes on average. A total of 34 levels were completed during this time, which constitute the basis for all conclusions drawn regarding balance score computation and personalized level generation. Evaluation of patient progress throughout the course of multiple sessions would've been ideal. This enables analysis of balance score consistency and evolution across time for each individual patient.

CHAPTER 7

Conclusion and future work

This thesis describes the analysis and development of a new SG for balance rehabilitation, which elevates the current state-of-the-art in difficulty adaptation mechanisms to accommodate player balance impairments. The SG design and game mechanics are based on requirements for SGs in balance rehabilitation, gathered from extensive literature research of related state-of-the-art work and interviews with physiotherapists. Prototype evaluation suggests that *Walk in the park* satisfyingly reflects the design criteria, discussed in section 4.1.1. As a result, impaired patients enjoyed *Walk in the park* as part of their rehabilitation schedule and would consider playing it again in the future. Furthermore, adaptation to level difficulty was deemed to adequately reflect patient impairment. To achieve this, levels are generated based on the requirements discussed in section 4.1.1 and incorporate the novel approach for quantifying balance ability described in this thesis. Based on these findings, it is evident that machine learning techniques applied to WBB data is capable of quantifying balance ability. Furthermore, evaluation results show that combining adequate balance estimation with findings from the second research objective provides personalized levels, properly reflecting patient balance impairments.

Despite promising results for evaluating balance ability from the evaluation phase, the conclusions are drawn based on a rather low number of participants and completed levels. This weakens expressiveness of the contributions developed in this thesis. Performing additional evaluation with a larger number of balance impaired participants could bring further insight into the relation between balance scoring and patient impairments. Additionally, performing multiple follow-up sessions with the same participants to evaluate progression over time and result consistency could prove beneficial.
As an extension to *Walk in the park*, further exercises for balance rehabilitation could be incorporated, such as standing exercises with varying stance, turning to look backwards or even standing on one leg. This would require customization for the level generation mechanism, because not every patient will be able to perform every exercise. With

support of experiences physiotherapists, a set of exercises that can be performed on the WBB could be compiled that target a more systematic balance rehabilitation. These new exercises would use the same approach for quantifying balance ability as discussed in this thesis. Supporting a variety of exercises could boost patient motivation and engagement due to a more versatile gaming experience and consequentially improve rehabilitation outcomes.

Even though balance metrics used to model healthy balance ability in this thesis are strong indicators for balance performance and postural control, the evaluation sessions have shown that balance impaired patients have considerably lower mean COP amplitude than healthy individuals. Even though literature suggests avoiding absolute balance metrics due to factors like incorrect foot positioning, additional metrics, such as the COP mean amplitude, have potential to improve model accuracy when used in combination with strict protocols for exercise execution. Another avenue for future research could be investigating further improvements in regards to quantifying balance ability by comparing model performance on a variety of feature configurations.

List of Figures

List of Tables

Glossary

API An Application Programming Interface defines a contract to access features or data of a system.. 16, 60, 61, 64, 74, 75, 76, 78, 80, 87, 89, 91, 93, 96, 104, 105, 107, 108

backend Part of an application that is not directly accessible to a user, responsible for data persistence and manipulation.. 49, 50, 58, 61, 73, 74, 75, 76, 80, 87, 92, 93, 96, 104, 107, 108, 110, 116, 146

BBS The Berg Balance Scale is a widely used exercise protocol for the evaluation of balance ability [26]. 12, 13, 18, 24, 33, 43, 44, 140, 149

Bluetooth Bluetooth is a wireless technology standard for data exchange across short distances using ultra high frequency radio waves.. 16

COP The center or pressure is an approximation of the body's center of pass projected verically onto the floor. It is widely used for standing balance assessment [37].. 15, 17, 18, 19, 20, 21, 22, 26, 36, 38, 39, 40, 41, 43, 44, 46, 47, 48, 54, 62, 68, 69, 70, 71, 74, 75, 77, 78, 80, 86, 91, 93, 94, 95, 96, 101, 102, 103, 104, 105, 109, 110, 112, 116, 117, 118, 123, 124, 125, 126, 128, 139, 140, 141, 144, 145

CRUD An acronym representing basic operations for persistent storage, refering to **C**reating new data, **R**eading data from storage, **U**pdating data and **D**eleting data.. 108

DTO A Data Transfer Object is a model class that is used for data exchange between systems.. 76, 84

exergame A serious game based on physical exertion. 1, 14, 35, 36, 44

framework Explicit design and implementation artifacts with well-defined boundaries to boost productivity throuh design and code reuse. [86]. 58, 61, 80

frontend User-facing part of an application which the user interacts with.. 49, 50, 58, 60, 73, 74, 76, 80, 82, 108

JSON JSON (JavaScript Object Notation) is a lightweight data-interchange format. [87]. 61, 76, 79, 84, 89, 108

SG A piece of software that combines a non-entertaining (serious) purpose with techniques from video game design [10]. 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 16, 21, 25, 26, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40, 46, 47, 49, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 73, 74, 80, 82, 83, 137, 138, 139, 141, 143, 149

URL The Uniform Resource Locator is a reference to the location of a resource within a computer network.. 75, 76, 107

WBB The Nintendo Wii Fit Balance Board is a force platform used as proprietary input device for the Nintendo Wii gaming console.. xi, xiii, 2, 3, 7, 14, 15, 16, 17, 18, 20, 21, 24, 26, 31, 35, 36, 37, 38, 39, 40, 41, 43, 44, 45, 46, 47, 51, 57, 58, 60, 62, 63, 64, 65, 66, 67, 69, 73, 74, 75, 79, 80, 82, 86, 87, 96, 102, 116, 132, 136, 137, 139, 143, 144, 145

Bibliography

[1] Michael W Kennedy, James P Schmiedeler, Charles R Crowell, Michael Villano, Aaron D Striegel, and Johan Kuitse. Enhanced feedback in balance rehabilitation using the nintendo wii balance board. In *2011 IEEE 13th international conference on e-health networking, applications and services*, pages 162–168. IEEE, 2011.

[2] Sarah F Tyson, Marie Hanley, Jay Chillala, Andrea Selley, and Raymond C Tallis. Balance disability after stroke. *Physical therapy*, 86(1):30–38, 2006.

[3] Gert Mertes, Greet Baldewijns, Pieter-Jan Dingenen, Tom Croonenborghs, and Bart Vanrumste. Automatic fall risk estimation using the nintendo wii balance board. In *HEALTHINF*, pages 75–81, 2015.

[4] James William Burke, MDJ McNeill, Darryl K Charles, Philip J Morrow, Jacqui H Crosbie, and Suzanne M McDonough. Optimising engagement for stroke rehabilitation using serious games. *The Visual Computer*, 25(12):1085, 2009.

[5] Fraser Anderson, Michelle Annett, and Walter F Bischof. Lean on wii: physical rehabilitation with virtual reality wii peripherals. *Stud Health Technol Inform*, 154(154):229–34, 2010.

[6] Ross A Clark, Adam L Bryant, Yonghao Pua, Paul McCrory, Kim Bennell, and Michael Hunt. Validity and reliability of the nintendo wii balance board for assessment of standing balance. *Gait & posture*, 31(3):307–310, 2010.

[7] José-Antonio Gil-Gómez, Roberto Lloréns, Mariano Alcañiz, and Carolina Colomer. Effectiveness of a wii balance board-based system (ebavir) for balance rehabilitation: a pilot randomized clinical trial in patients with acquired brain injury. *Journal of neuroengineering and rehabilitation*, 8(1):1–10, 2011.

[8] Nunzio Alberto Borghese, Michele Pirovano, Pier Luca Lanzi, Seline Wüest, and Eling D de Bruin. Computational intelligence and game design for effective at-home stroke rehabilitation. *Games for Health: Research, Development, and Clinical Applications*, 2(2):81–88, 2013.

[9] Davide Bacciu, Stefano Chessa, Claudio Gallicchio, Alessio Micheli, Luca Pedrelli, Erina Ferro, Luigi Fortunati, Davide La Rosa, Filippo Palumbo, Federico Vozzi, et al.

A learning system for automatic berg balance scale score estimation. *Engineering Applications of Artificial Intelligence*, 66:60–74, 2017.

[10] Damien Djaouti, Julian Alvarez, and Jean-Pierre Jessel. Classifying serious games: the g/p/s model. In *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*, pages 118–136. IGI Global, 2011.

[11] Stefan Göbel, Sandro Hardy, Viktor Wendel, Florian Mehm, and Ralf Steinmetz. Serious games for health: personalized exergames. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1663–1666. ACM, 2010.

[12] Fedwa Laamarti, Mohamad Eid, and Abdulmotaleb El Saddik. An overview of serious games. *International Journal of Computer Games Technology*, 2014:11, 2014.

[13] Richard N Landers and Rachel C Callan. Casual social games as serious games: The psychology of gamification in undergraduate education and employee training. In *Serious games and edutainment applications*, pages 399–423. Springer, 2011.

[14] Ramon Hervas, David Ruiz-Carrasco, Tania Mondejar, and Jose Bravo. Gamification mechanics for behavioral change: a systematic review and proposed taxonomy. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 395–404, 2017.

[15] Aleksandra E Olszewski and Traci A Wolbrink. Serious gaming in medical education: a proposed structured framework for game development. *Simulation in Healthcare*, 12(4):240–253, 2017.

[16] Timothy M Dall, Paul D Gallo, Ritasree Chakrabarti, Terry West, April P Semilla, and Michael V Storm. An aging population and growing disease burden will require alarge and specialized health care workforce by 2025. *Health affairs*, 32(11):2013–2020, 2013.

[17] Pakaratee Chaiyawat and Kongkiat Kulkantrakorn. Effectiveness of home rehabilitation program for ischemic stroke upon disability and quality of life: a randomized controlled trial. *Clinical neurology and neurosurgery*, 114(7):866–870, 2012.

[18] Xian-Liang Liu, Jing-Yu Tan, Tao Wang, Qi Zhang, Min Zhang, Li-Qun Yao, and Jin-Xiu Chen. Effectiveness of home-based pulmonary rehabilitation for patients with chronic obstructive pulmonary disease: a meta-analysis of randomized controlled trials. *Rehabilitation Nursing*, 39(1):36–59, 2014.

[19] Anargyros Chatzitofis, David Monaghan, Edmond Mitchell, Freddie Honohan, Dimitrios Zarpalas, Noel E O'Connor, and Petros Daras. Hearthealth: a cardiovascular disease home-based rehabilitation system. *Procedia Computer Science*, 63:340–347, 2015.

[20] Valeria Manera, Pierre-David Petit, Alexandre Derreumaux, Ivan Orvieto, Matteo Romagnoli, Graham Lyttle, Renaud David, and Philippe H Robert. 'kitchen and cooking,'a serious game for mild cognitive impairment and alzheimer's disease: a pilot study. *Frontiers in aging neuroscience*, 7:24, 2015.

[21] Ben Sawyer. From cells to cell processors: the integration of health and video games. *IEEE computer graphics and applications*, 28(6):83–85, 2008.

[22] Minhua Ma and Kamal Bechkoum. Serious games for movement therapy after stroke. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 1872–1877. IEEE, 2008.

[23] MARCELLA Caglio, L Latini-Corazzini, Federico D'Agata, Franco Cauda, Katiuscia Sacco, S Monteverdi, M Zettin, S Duca, and G Geminiani. Virtual navigation for memory rehabilitation in a traumatic brain injured patient. *Neurocase*, 18(2):123–131, 2012.

[24] Francesco Ricciardi and Lucio Tommaso De Paolis. A comprehensive review of serious games in health professions. *International Journal of Computer Games Technology*, 2014:9, 2014.

[25] Biel Moyà Alcover, Antoni Jaume-i Capó, Javier Varona, Pau Martinez-Bueso, and Alejandro Mesejo Chiong. Use of serious games for motivational balance rehabilitation of cerebral palsy patients. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 297–298, 2011.

[26] Susan W Muir, Katherine Berg, Bert Chesworth, and Mark Speechley. Use of the berg balance scale for predicting multiple falls in community-dwelling elderly people: a prospective study. *Physical therapy*, 88(4):449–459, 2008.

[27] Lisa Blum and Nicol Korner-Bitensky. Usefulness of the berg balance scale in stroke rehabilitation: a systematic review. *Physical therapy*, 88(5):559–566, 2008.

[28] Jun Young Han, Jong Moon Kim, Shin Kyoung Kim, Jin Sang Chung, Hyun-Cheol Lee, Jae Kuk Lim, Jiwon Lee, and Kawn Yong Park. Therapeutic effects of mechanical horseback riding on gait and balance ability in stroke patients. *Annals of rehabilitation medicine*, 36(6):762, 2012.

[29] Scott Bennie, Kathryn Bruner, Allan Dizon, Holly Fritz, Bob Goodman, and Saundra Peterson. Measurements of balance: comparison of the timed" up and go" test and functional reach test with the berg balance scale. *Journal of Physical Therapy Science*, 15(2):93–97, 2003.

[30] Johanna Jonsdottir and Davide Cattaneo. Reliability and validity of the dynamic gait index in persons with chronic stroke. *Archives of physical medicine and rehabilitation*, 88(11):1410–1415, 2007.

[31] Antoni Jaume-i Capó, Pau Martínez-Bueso, Biel Moyà-Alcover, and Javier Varona. Interactive rehabilitation system for improvement of balance therapies in people with cerebral palsy. *IEEE transactions on neural systems and rehabilitation engineering*, 22(2):419–427, 2013.

[32] Simon Brassard, Martin J-D Otis, Alexandre Poirier, and Bob-Antoine J Menelas. Towards an automatic version of the berg balance scale test through a serious game. In *Proceedings of the Second ACM Workshop on Mobile Systems, Applications, and Services for Healthcare*, page 5. ACM, 2012.

[33] René Baranyi, Rainer Willinger, Nadja Lederer, Thomas Grechenig, and Wolfgang Schramm. Chances for serious games in rehabilitation of stroke patients on the example of utilizing the wii fit balance board. In *2013 IEEE 2nd International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–7. IEEE, 2013.

[34] Letizia Castelli, Luca Stocchi, Maurizio Patrignani, Giovanni Sellitto, Manuela Giuliani, and Luca Prosperini. We-measure: toward a low-cost portable posturography for patients with multiple sclerosis using the commercial wii balance board. *Journal of the neurological sciences*, 359(1-2):440–444, 2015.

[35] Avril Mansfield and Elizabeth L Inness. Force plate assessment of quiet standing balance control: Perspectives on clinical application within stroke rehabilitation. *Rehabilitation Process and Outcome*, 4:RPO–S20363, 2015.

[36] Yocheved Laufer, R Schwarzmann, D Sivan, and E Sprecher. Postural control of patients with hemiparesis: force plates measurements based on the clinical sensory organization test. *Physiotherapy theory and practice*, 21(3):163–171, 2005.

[37] Daniel J Goble, Brian L Cone, and Brett W Fling. Using the wii fit as a tool for balance assessment and neurorehabilitation: the first half decade of "wii-search". *Journal of neuroengineering and rehabilitation*, 11(1):12, 2014.

[38] Jeffrey D Holmes, Mary E Jenkins, Andrew M Johnson, Michael A Hunt, and Ross A Clark. Validity of the nintendo wii® balance board for the assessment of standing balance in parkinson's disease. *Clinical Rehabilitation*, 27(4):361–366, 2013.

[39] Harrison L Bartlett, Lena H Ting, and Jeffrey T Bingham. Accuracy of force and center of pressure measures of the wii balance board. *Gait & posture*, 39(1):224–228, 2014.

[40] WiiBrew contributors. Wiimote/library - wiibrew. `https://wiibrew.org/wiki/Wiimote/Library`, 2015. [Online; accessed 25-December-2020].

[41] Giacomo Severini, Sofia Straudi, Claudia Pavarelli, Marco Da Roit, Carlotta Martinuzzi, Laura Di Marco Pizzongolo, and Nino Basaglia. Use of nintendo wii balance board for posturographic analysis of multiple sclerosis patients with minimal balance impairment. *Journal of neuroengineering and rehabilitation*, 14(1):19, 2017.

[42] Tian Bao, Brooke N Klatt, Susan L Whitney, Kathleen H Sienko, and Jenna Wiens. Automatically evaluating balance: a machine learning approach. *IEEE transactions on neural systems and rehabilitation engineering*, 27(2):179–186, 2019.

[43] Riann M Palmieri, Christopher D Ingersoll, Marcus B Stone, and B Andrew Krause. Center-of-pressure parameters used in the assessment of postural control. *Journal of sport rehabilitation*, 11(1):51–66, 2002.

[44] Fay B Horak. Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls? *Age and ageing*, 35(suppl_2):ii7–ii11, 2006.

[45] Issam El Naqa and Martin J Murphy. What is machine learning? In *Machine Learning in Radiation Oncology*, pages 3–11. Springer, 2015.

[46] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[47] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.

[48] Hannes Hapke and Catherine Nelson. *Building Machine Learning Pipelines*. O'Reilly Media, Inc., 2020.

[49] Thamindu Dilshan Jayawickrama. Introduction to anomaly detection | by thamindu dilshan jayawickrama. `https://towardsdatascience.com/introduction-to-anomaly-detection-c651f38ccc32`, 2020. [Online; accessed 11-January-2021].

[50] Itziar Irigoien, Basilio Sierra, and Concepción Arenas. Towards application of one-class classification methods to medical data. *The Scientific World Journal*, 2014, 2014.

[51] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46, 2000.

[52] Martin Glinz and Roel J Wieringa. Guest editors' introduction: Stakeholders in requirements engineering. *IEEE software*, 24(2):18–20, 2007.

[53] Elsa Marcelino-Jesus, Joao Sarraipa, Carlos Agostinho, and Ricardo Jardim-Goncalves. A requirements engineering methodology for technological innovations assessment. In *ISPE CE*, pages 577–586, 2014.

[54] Olatunji J Okesola, Kennedy O Okokpujie, Rowland Goddy-Worlu, Afolakemi Ogunbanwo, and Iheanetu Olamma. Qualitative comparisons of elicitation techniques in requirement engineering. *QUALITATIVE COMPARISONS OF ELICITATION TECHNIQUES IN REQUIREMENT ENGINEERING*, 14(2):565–570, 2019.

[55] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*, volume 5. Springer Science & Business Media, 2012.

[56] Masooma Yousuf and M Asger. Comparison of various requirements elicitation techniques. *International Journal of Computer Applications*, 116(4), 2015.

[57] William Young, Stuart Ferguson, Sebastien Brault, and Cathy Craig. Assessing and training standing balance in older adults: a novel approach using the 'nintendo wii'balance board. *Gait & posture*, 33(2):303–305, 2011.

[58] Paul Krebs, Jack E Burkhalter, Bert Snow, Jeff Fiske, and Jamie S Ostroff. Development and alpha testing of quitit: an interactive video game to enhance skills for coping with smoking urges. *JMIR research protocols*, 2(2):e35, 2013.

[59] Tiago Gomes, Tiago Abade, José C Campos, Michael Harrison, and José Luís Silva. Rapid development of first person serious games using the apex platform: The asthma game. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 169–174. ACM, 2014.

[60] Stefania Artioli, Riccardo Berta, Alessandro De Gloria, Andrea Pomicino, and Nicola Secco. A serious game to inform about hiv prevention: Hinvaders, a case study. In *Games for Health*, pages 3–13. Springer, 2013.

[61] Mitra Memarzia and Kam Star. Choices and voices–a serious game for preventing violent extremism. In *Intelligence management*, pages 133–142. Springer, 2011.

[62] René Baranyi, Bernhard Steyrer, Lukas Lechner, Gevher N Agbektas, Nadja Lederer, and Thomas Grechenig. Nutritionrush-a serious game to support people with the awareness of their nutrition intake. In *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8. IEEE, 2017.

[63] D Parvati, WL Heinrichs, and Y Patricia. Clinispace: a multiperson 3d online immersive training environment accessible through a browser. *Medicine Meets Virtual Reality 18: NextMed*, 163:173, 2011.

[64] Jing Qin, Yim-Pan Chui, Wai-Man Pang, Kup-Sze Choi, and Pheng-Ann Heng. Learning blood management in orthopedic surgery through gameplay. *IEEE computer graphics and applications*, 30(2):45–57, 2009.

[65] Kathrin Maria Gerling, Jonas Schild, and Maic Masuch. Exergame design for elderly users: the case study of silverbalance. In *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, pages 66–69. ACM, 2010.

[66] Belinda Lange, Sheryl Flynn, Rachel Proffitt, Chien-Yen Chang, and Albert "Skip" Rizzo. Development of an interactive game-based rehabilitation tool for dynamic balance training. *Topics in stroke rehabilitation*, 17(5):345–352, 2010.

[67] https://commons.wikimedia.org/w/index.php?curid=11523156 Ennepetaler86 Eigenes Werk, CC BY 3.0. Support vector machine - wikipedia. https://de.wikipedia.org/wiki/Support_Vector_Machine, 2020. [Online; accessed 02-January-2021].

[68] Saleh Alaliyat. Video-based fall detection in elderly's houses. Master's thesis, 2008.

[69] Maarit Piirtola and Pertti Era. Force platform measurements as predictors of falls among older people–a review. *Gerontology*, 52(1):1–16, 2006.

[70] Itshak Melzer, N Benjuya, and Jacob Kaplanski. Postural stability in the elderly: a comparison between fallers and non-fallers. *Age and ageing*, 33(6):602–607, 2004.

[71] Google LLC. Google scholar. https://scholar.google.com, 2021. [Online; accessed 10-January-2021].

[72] Michael Gusenbauer. Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics*, 118(1):177–214, 2019.

[73] Belinda Lange, Chien-Yen Chang, Evan Suma, Bradley Newman, Albert Skip Rizzo, and Mark Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1831–1834. IEEE, 2011.

[74] Fabrizia Corona, Alex De Vita, Giovanni Filocamo, Michaela Foa, Pier Luca Lanzi, Amalia Lopopolo, and Antonella Petaccia. Lower limb rehabilitation in juvenile idiopathic arthritis using serious games. *arXiv preprint arXiv:2006.02187*, 2020.

[75] F Noveletto, AV Soares, BA Mello, CN Sevegnani, FLF Eichinger, M Da S Hounsell, and P Bertemes-Filho. Biomedical serious game system for balance rehabilitation of hemiparetic stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(11):2179–2188, 2018.

[76] Inc. MongoDB. What is mongodb? | mongodb. https://www.mongodb.com/what-is-mongodb, 2021. [Online; accessed 08-January-2021].

[77] Guido Van Rossum et al. Python programming language. In *USENIX annual technical conference*, volume 41, page 36, 2007.

[78] Adobe Inc. Ui/ux-design und zusammenarbeit | adobe xd. https://www.adobe.com/at/products/xd.html, 2021. [Online; accessed 10-January-2021].

[79] JetBrains s.r.o. Intellij idea: The capable and ergonomic java ide by jetbrains. https://www.jetbrains.com/idea/, 2021. [Online; accessed 20-January-2021].

[80] Mario Zechner. libgdx. https://libgdx.badlogicgames.com/features.html, 2021. [Online; accessed 18-January-2021].

[81] Gradle Inc. Gradle build tool. https://gradle.org/, 2021. [Online; accessed 19-January-2021].

[82] Michael Diamond. micromu/wiiremotej: Wiiremotej mirror, wiimote java library written by michael diamond. https://github.com/micromu/WiiRemoteJ, 2012. [Online; accessed 05-January-2021].

[83] BlueCove Team. Bluecove - bluecove jsr-82 project. http://bluecove.org/, 2008. [Online; accessed 19-January-2021].

[84] Inc. MongoDB. mongod - mongodb manual. https://docs.mongodb.com/manual/reference/program/mongod/#mongodb-binary-bin.mongod, 2021. [Online; accessed 10-April-2021].

[85] Michela Massi. Schema of an autoencoder. https://commons.wikimedia.org/wiki/File:Autoencoder_schema.png, 2019. [Online; accessed 10-February-2022].

[86] Dirk Riehle. *Framework design: A role modeling approach*. PhD thesis, ETH Zurich, 2000.

[87] json.org. Json. https://www.json.org/json-en.html, 2021. [Online; accessed 08-January-2021].

Appendix

Path generation heuristic

Algorithm A.1: Path generation heuristic (*generate* method located in *PathGenerator*)

Input: An instance of *PathGeneratorParams*
Output: A list of *Nodes* representing the found path

```
1  for i ← 1 to 100 do
2      reset internal counters left_, right_, up_ and down_
3      path.clear()
4      grid = Node[gridSize][gridSize] // initialize empty Node matrix
5      position = Point(gridSize/2, gridSize/2)
6      while !pathFound() && direction != null do
7          currentNode = grid[position.x][position.y]
8          direction = getNextDirection(currentNode)
9          if direction != null then
10             currentNode.direction = direction
11             currentNode.visited = true
12             path.add(currentNode)
13             decrease number of nodes needed for direction
14         end
15     end
16     if pathFound() then
17         break for;
18     end
19 end
20 return path
```

Experience score calculation

Algorithm A.2: Experience score calculation

Input: Current and historic score for a specific direction
Output: Experience point gain or decrease for direction

```
improvement = currentScore / historicScore
if improvement < 1 then
    xpGain = -(30 - computeXp(currentScore)) * improvement
else
    xpGain = computeXp(currentScore)
end
return xpGain
```

Generation of valid path configurations

Algorithm A.3: Generation of valid paths parameters in *PathParamsGenerator*

Input: An instance of *PathGeneratorParams* containing the targetted *gridSize*
Output: A list of valid *PathGeneratorParams*

```
1  validPathGeneratorParams = emptyList()
2  totalNumPathTiles = 2 * params.gridSize - 6
3  for l ← 0 to totalNumPathTiles - 1 do
4      params.numLeftTiles = l
5      for r ← 0 to totalNumPathTiles - 1 do
6          params.numRightTiles = r
7          for u ← 0 to totalNumPathTiles - 1 do
8              params.numUpTiles = u
9              for d ← 0 to totalNumPathTiles - 1 do
10                 params.numDownTiles = d
11                 if params.getTotalNumTiles() == totalNumPathTiles then
12                     pathGenerator.generate(params)
13                     if pathGenerator.pathGeneratedSuccessfully() then
14                         validPathGeneratorParams.add(params)
15                     end
16                 end
17             end
18         end
19     end
20 end
21 return validPathGeneratorParams
```

Conversion of evaluation result to path configuration

Algorithm A.4: Conversion of *EvaluationResult* to path configuration

Input: An instance of *EvaluationResult* describing the patient's performance executing weight shift exercises during gameplay

Output: *PathGeneratorParams* that describe an optimal path for the given *EvaluationResult*

```
1  totalNumPathTiles = 2 * gridSize - 6
2  left, right, up, down = 0
3  while sum(left, right, up, down) < totalNumPathLength do
4  |  leftRatio = evaluationResult.left / (totalNumPathTiles - left)
5  |  ... rightRatio, upRatio and downRatio computed analogously
6  |  minRatio = min(leftRatio, rightRatio, upRatio, downRatio)
7  |  if minRatio == leftRatio then
8  |  |  left++
9  |  end
10 |  ... right, up and down incremented in analogous fashion
11 end
12 return PathGeneratorParams(gridSize, left, right, up, down)
```

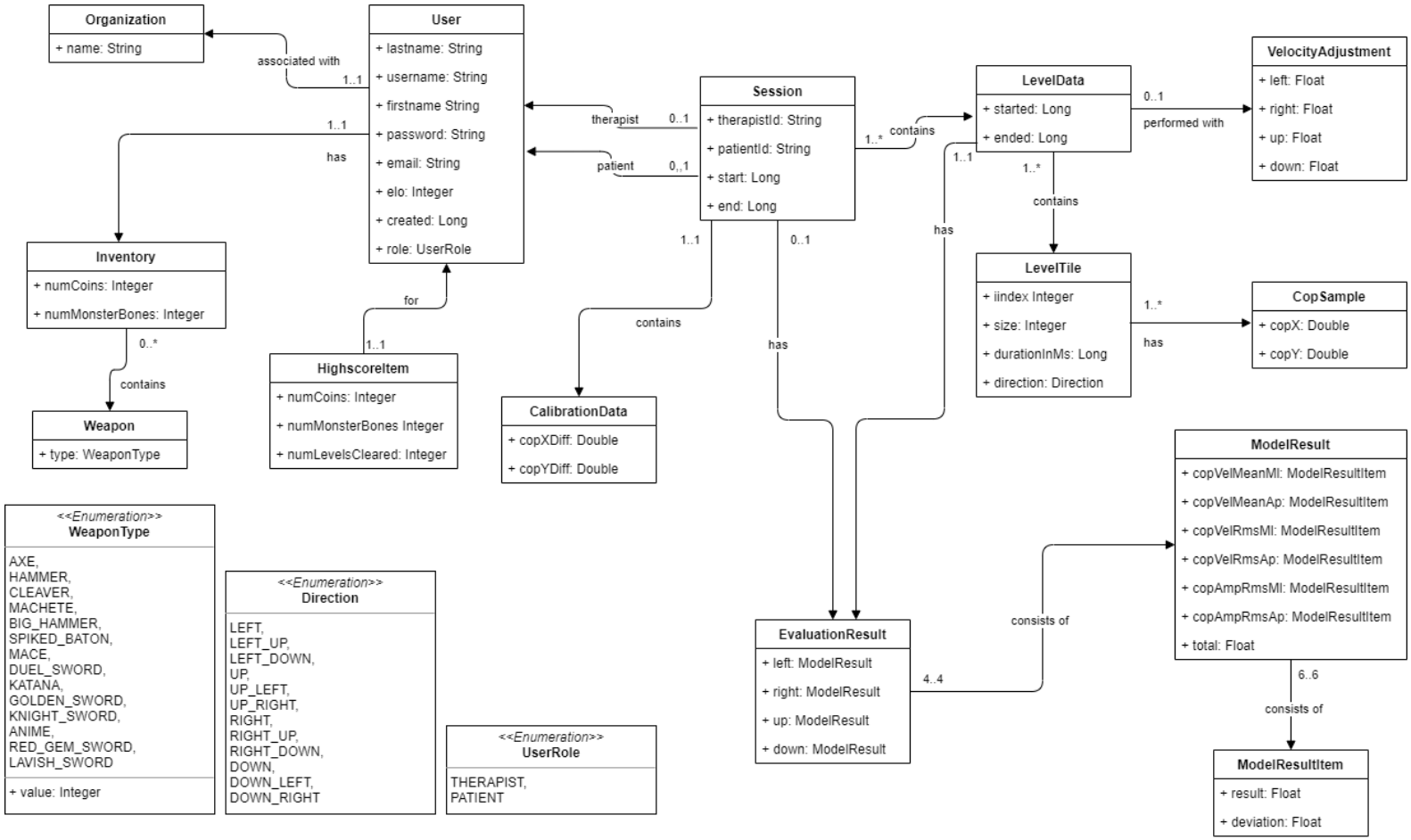


Figure A1: Data model of *Walk in the park*

Questionnaire for physiotherapists

Fragebogen Therapeut	Stimmt völlig zu	Stimmt meist zu	Stimmt teilweise zu	Stimmt kaum zu	Stimmt nicht zu
Levelgenerierung					
L1) Die von der Anwendung berechnete Evaluierung des Bewegungsverhaltens spiegelt die Beeinträchtigungen hinsichtlich Balancefähigkeit des Patienten wieder.					
L2) Die von der Anwendung generierten Levels passen sich auf die Bedürfnisse des Patienten an (Forcierung in schwächste Richtung).					
L3) Die Anwendung passt den Schwierigkeitsgrad (Länge) der generierten Levels an das Leistungsniveau des Patienten in hinreichender Weise an.					
L4) Die Anwendung führt den Patienten schrittweise an sein Leistungsniveau heran.					
Spielprinzip					
S1) Die Spielmechaniken sind einfach gehalten und der Patient war nicht überfordert.					
S2) Der Fortschritt des Patienten hinsichtlich Performance während der Sitzung ist gut ersichtlich.					
S4) Es gibt keine wahrnehmbare Zeitverzögerung zwischen einer Gewichtsverlagerung auf dem Balance Board und einer Bewegung im Spiel.					
S4) Die Übung (Gewichtsverlagerung) auf dem Balance Board ist eine wirkungsvolle Übung zur Balance-Rehabilitation.					
S5) Ich könnte mir vorstellen das Spiel als zusätzliche Maßnahme zur Balance-Reha weiterhin einzusetzen.					

Figure A2: Original questionnaire handed to the physiotherapists for prototype evaluation

Questionnaire for patients

Fragebogen Patient	Stimmt völlig zu	Stimmt meist zu	Stimmt teilweise zu	Stimmt kaum zu	Stimmt nicht
P1) Das Spiel war zu anstrengend und hat sich hinsichtlich Schwierigkeitsgrad nicht auf mich angepasst.					
P2) Die absolvierte Übung (Gewichtsverlagerung) ist zu einfach.					
P3) Ich würde mich darüber freuen, wenn das Spiel in meiner Therapie häufiger eingesetzt werden würde.					
P4) Ich war motiviert noch mehr zu spielen.					
P5) Das Spiel hat mich frustriert.					
P6) Ich würde das Spiel auch zuhause spielen.					
P7) Es war schwer dem Spielprinzip zu folgen.					
P8) Das Feedback nach jedem Level und am Ende der Sitzung hat mir geholfen meine Schwächen hinsichtlich Balancefähigkeit besser zu verstehen.					
P9) Die Bewertung meiner Balancefähigkeit war treffend.					
P10) Das Spiel war anstrengend, jedoch machbar.					
P11) Ich wurde langsam an meine Leistungsgrenze herangeführt.					

Figure A3: Original questionnaire handed to the patients for prototype evaluation