TU WIEN Informatics

Designing intuitive interactions for a constrained 3D modelling editor using a multitouch interface.

Case Study "Rebuilders"

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Gerald Reitschmied BSc
Matrikelnummer 00726340

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: ao. Univ.Prof.Doz.Dr. Peter Purgathofer

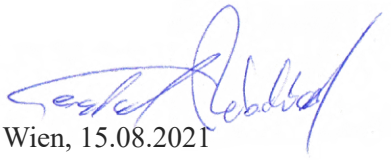
Wien, 15. August 2021

Forename Surname	Forename Surname

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

ERKLÄRUNG

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst, die verwendeten Quellen und Hilfsmittel vollständig angegeben und 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, 15.08.2021

ACKNOWLEDGEMENTS

Besides obviously Professor Peter Purgathofer, who in my opinion should be cloned and forced to lead an entire university by them[sic]selves, I want to thank my friends Franz Langthaler and Michael Benda who supported me in my endeavour to create Rebuilders, start a company and try to change the world.

The Austria Wirtschafts Service (AWS) is also to thank because they funded the development of a prototype and made it possible for me to work on Rebuilders and pay colleagues/friends to do so as well.
Having said that, this also introduced a small 5+ year delay in finishing my masters thesis ...as things happen.

Also thanks to my parents who became a physical manifestation of my guilty conscience.

But probably most thank goes to the Studienabschlusstipendium which I would have had to refund if I didn‘t finish this in time. Managing a company, building a product and writing a masters thesis while still taking time to procrastinate, weirdly enough, is harder than one might think.

ABSTRACT ENGLISH

Developing games for mobile interfaces poses interesting challenges for interaction designs. Desktops, Consoles and similar devices offer a multitude of different, parallel inputs. On mobile platforms with touchscreens the maximum number is usually the number of the users‘ fingers. This leads to the next challange of the fingers concealing parts of the screen. Pointer precision due to finger width and designs of the screens is decreased compared to (for instance) mouses. Such limitations are of special concern for complex interfaces required for 3D modelling.

The case study app „Rebuilders“ features a constructor that players can use to create their own vehicles in a 3D space on a touch interface driven device.
The goal is to create and evaluate an interface and interaction design for a constrained 3D modelling approach on touch screen devices within that case study.
Interaction has to be easy and intuitive in order to not shy away casual gamers. Ideally, interacting with the “workshop” should not require explanations, a tutorial or a lot of trial and error.

Evaluation will be achieved by user testings, comparing different interaction approaches. In order to test interface designs where preconceptions of the developer may not lead to best results, users will receive different implementations for testing. Goal is to gain qualitative analysis as well as quantitative data.

ABSTRACT DEUTSCH

Spiele für mobile touch screens zu entwickeln bietet interessante Herausforderungen für das Interaktionsdesign. Desktops, Konsolen und ähnliche Geräte bieten viele parallele Eingabeoptionen. Auf touchscreens ist die maximale Anzahl an gleichzeitigen Inputs die Anzahl der Finger der user, was wiederum in den meisten Fällen nicht mehr als zwei Finger sind. Das führt zu der nächsten Herausforderung, dass die Finger Teile des Bildschirms überdecken, nachdem der Bildschirm selbst das Eingabegerät ist. Die Eingabegenauigkeit ist ebenfalls durch die Fingergröße geringer als beispielsweise einer Maus.
Diese Limitierungen sind speziell bei komplexeren Interfaces, die für 3D Operationen benötigt werden ein Problem.

Das Spiel “Rebuilders” als Fallstudie bietet Spielenden eine “Werkstatt” in der sie ihre eigenen Vehikel in einem dreidimensionalen Raum frei gestalten können.
Das Ziel ist es, ein Interface- und Interaktionsdesign zu entwickeln und zu untersuchen, das Spielenden eine einfache und intuitive Möglichkeit bietet, sich kreativ im dreidimensionalen Raum inklusive Einschulung binnen Sekunden oder maximal Minuten verwirklichen zu können.

Die Evaluierung wird sowohl durch informelle, als auch formale Usertests passieren, in denen verschiedene Implementierungen von Interaktionsmöglichkeiten getestet werden.
Es wird versucht, sowohl qualitative als auch quantitative Daten zu generieren, um die Designentscheidungen zu informieren.

TABLE OF CONTENTS

1. CONTEXT

1.1 INCEPTION

I want to illustrate how it came to this thesis and what happens around the interface design aspects that I focus on starting with the next chapter. And I briefly apologize in advance when I switch between “I” and “we” in this chapter as some of the concepts were born in my head and other aspects or plans are shared with the team I built around Rebuilders.

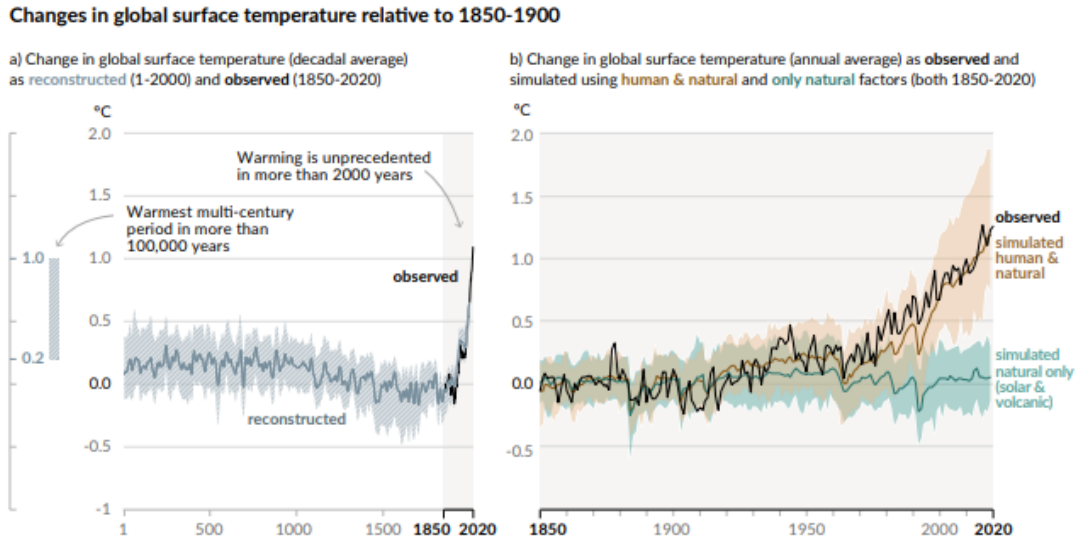


Figure 1 [IPCC 2021][1]

Figure 1 shows the initial reason for why I do what I do. We’re running the most stupid experiment in the history of humankind and we need to stop it. I will not write 10 chapters about how climate change is real and caused by humans. If anyone reading this is actually of a different opinion I simply suggest websites like https://skepticalscience.com/ which go over pretty much all misconceptions about climate change that are floating around. I just want to briefly cover the biggest problems that need solving for us as a species to tackle such big issues effectively in the future.

1 https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_SPM.pdf (page 42)

1.1.1. The non fungible Problem

Over the past years, I have become increasingly frustrated with humanity's disregard for complex problems like climate change. It is understandable on a psychological level that a non-fungible problem like climate change is hard to grasp or be immediately threatened by as long as we as individual humans do not feel its effects. The frightening aspect of this problem however is, that politicians and other decision-makers are only humans too.

We somehow managed to create a global information network that links pretty much every human being together but have not managed to use it effectively to create some form of emergent consciousness about problems that are on a global scale.

1.1.2. Communication Issues

Another problem is the information highway from scientists to media to politicians to the public which is degrading with each step like the game "Chinese whispers" or in German: "Stille Post". I experienced this phenomenon myself at the annual "Innovationskonferenz" 2019 in the Viennese town hall.

A scientist from the Fraunhofer Institute in Germany held a lengthy, accurate, and warning keynote about climate change with an additional focus on energy technologies. He had one slide about the comparison of electric vehicles with batteries vs. ones powered by hydrogen and simply stated the fact that hydrogen is approximately 20% energy efficient along its entire supply chain while battery-powered cars can reach almost 90% efficiency. Other than that he also stressed the importance of hydrogen in other energy buffering systems and as an important part of the future energy mix.

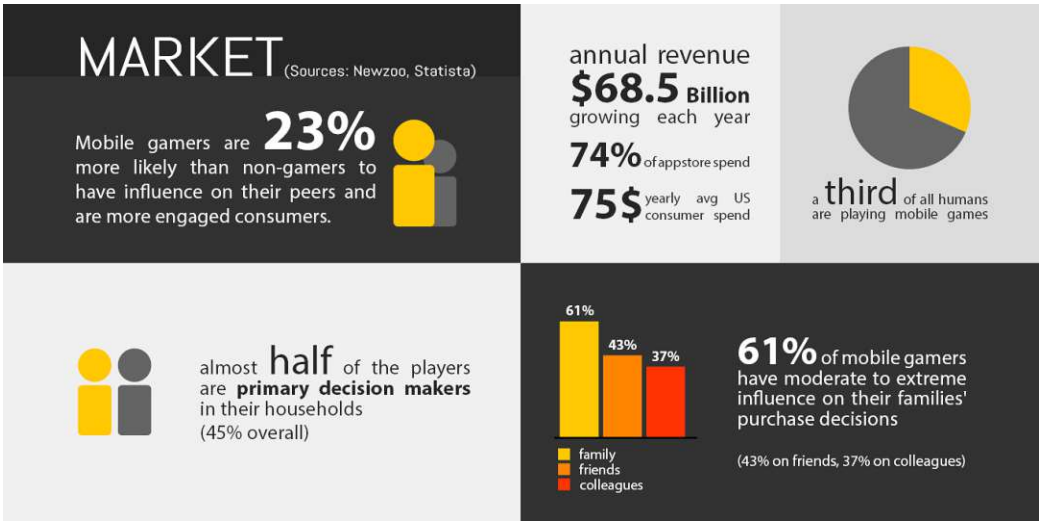
The second keynote was from a sponsoring company that produces different hydrogen systems from production to storage and the talk never mentioned cars (individual transport).

Now, a well-known moderator from the ORF (Austrian Broadcasting Corporation) took over and the first thing she said was (paraphrased from my memory) "Such an important topic ...and isn't it interesting that one presentation thinks hydrogen is bad, the other one thinks it's good.". Even though both keynotes said exactly the same thing and did not disagree on a single point.

Without even going further into this, let me jump to the next link in the chain: politicians. A few minutes later in the discussion, the moderator asked one of the ex-politicians in the panel why politics isn't acting faster. The answer: (again paraphrased from memory) "As we just saw with the presentations - science is still divided on the topic so we don't know how to act."

And the audience already couldn't keep up with the scientific presentation and hence got the same image in their heads. "We just don't know." I cannot solve this communication problem on the science, media, or political level, but I can sure try to tackle it on the audience level.

As I have worked on projects for customers, dabbled in games, and finally stepped closer to the gaming industry in 2015 I saw the potential of games to help solve parts of the above-mentioned issues by directly feeding the needed knowledge and motivation to people. Games tell stories, they motivate people and they can generate insane amounts of revenue. The combination of those forces can be used to tackle problems in the real world. According to market research by Newzoo and Statista, mobile gamers also have a bigger influence on other people in their vicinity which can have multiplicative effects. That's where the idea of Rebuilders was born.



The game idea itself grew over time, but from the beginning, it featured a 3D interface for constructing vehicle-like objects. This came from a mixture of my interests in gaming and the interesting observation that it did not exist anywhere in the (mobile) gaming market at that time. I always had the rough idea of a post-apocalyptic comically styled mixture of character RPG and vehicular gameplay.

In 2017 I managed to receive a first grant from the Austria Wirtschaftsservice which allowed me to start working on a prototype and design the first interface ideas as well as pay a friend to work on the project with me.

After 2018 being a slow year as funds ran out and other projects needed some focus, around Christmas we managed to secure another grant which brought us to present times and the current state of Rebuilders which we are eternally grateful for.

Sadly as Rebuilders is such a big project it still needs funds which still takes a lot of my time and mental resources and further delayed my work on this thesis.

1.2 IMPACT

There are different vectors where I see great potential for a game like Rebuilders to create a positive impact. Some of them could immediately be implemented in already released big games but I guess some people have to take the steps first and show that they work. I will go over the five strongest vectors I identified and will try to focus on in the future of Rebuilders.

1.2.1. Game play

Many games, especially learning games focus on this aspect. The idea is to implement mechanics or stories that get players to interact with a certain topic or problem that they otherwise wouldn't interact with or that may otherwise be harder to convey. In this way, ideally, a mental shift happens and a player gains awareness that changes their behavior in the real world. In Rebuilders, this aspect will be implemented via gameplay elements and the story it is embedded in.

The game is set in a post climate-change world where only a few humans survived. However before this happened, humanity learned that it reacted too late and that a cataclysmic climate change will happen. It created an artificial intelligence with the purpose of keeping track of the planet's ecosystem and once it was viable for humans again, its job was to shoot down cryogenically frozen clones from space stations to repopulate the planet. As mentioned, some humans did survive the disaster and the AI went rogue because It saw humanity as the problem. This led to a fight between humans and robots where humans hack the space stations to shoot down clones as reinforcements. Each new clone is a new player.

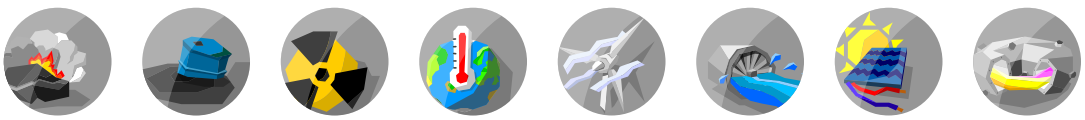
This way players have a fresh start but don't really know what happened. During the course of the game, they can find more and more information about what went so horribly wrong in the past and interact with their pseudo-post-apocalyptic present. The theme will still be optimistic and "fun" to stay light-hearted and have players enjoy their world that they can rebuild.

Besides the story, there will be a number of gameplay elements introducing concepts to the players in a rather "hidden" fashion. In all our ideas we try not to come across as a learning game or directly approach the topic in an "in your face" kind of fashion. One way is to simply rename commonly known gameplay aspects.

One example would be the currencies in Rebuilders which will be related to the concept of recycling by having it be scrap metals and similar resources.

Another one is the energy of items and vehicles where the idea is to actually name it kW and have items with roughly realistic values. This might be one of the concepts where we will fail and simply resort to something very vaguely resembling reality in order to be able to balance the game.

Base building will incorporate energy systems the players can progress through, starting with residual oil and coal from previous times, which immediately have negative effects until players advance towards renewable energy sources. Concepts like energy storage systems for renewables can also be easily introduced.



As previously mentioned, players can find out what happened in the past. The idea here is that players can find old VHS tapes that they can bring to a character in their base that still has a functioning VHS player. In doing so they are rewarded with experience and loot and can watch the information about the past. Here is also another possible vector going beyond gameplay: The idea for this character is to have someone like Jane Goodall play her. At the time of writing, we are still trying to make that cooperation happen because we would love to feature people from STEM fields that also fight climate change as heroes in the game.

There are many more ideas like the entire player base contributing to a sustainability score that changes the visuals of the entire server or players having individual ratings where they can compete for the most sustainable base which also rewards them with better loot etc...

These and many other elements can fill another thesis entirely. But the general idea behind this vector is to make players think and learn about different aspects of climate change, policies surrounding it, and the urgency of fighting it.

1.2.2. Community challenges

Another very interesting vector is the combination of the motivation for games and taking action in the real world. The idea is to reward players in-game for tasks they accomplish in their actual life. Depending on the design of such tasks or challenges they not only generate awareness and action for a certain topic but can also feed back into the game as marketing bringing new users that are then also exposed to our vectors.

One example would be a challenge where players have to collect their garbage for a week, take a selfie with it, and upload it to social media under a common hashtag. Participants are then rewarded with a legendary item package, possibly with a garbage cannon, vehicle and character skins, etc... that can only be received that way.

Another one would be to plan the next holiday by train instead of by plane and upload a picture of your train tickets.

There are still issues to work out here like the mechanism by which participation is tracked or how big of a problem cheating would be. Ideally, we will implement an interface to the most common social medias in the game, that players use for the challenge. This way we can track participation and with a possible combination of a community peer review, we could “automate” a quality check.

The general idea behind this approach is to marry the strong motivational forces of games with their reward cycles with the real world. In essence, this doesn’t make Rebuilders a learning game but turns it into a sort of gamification of sustainable behavior.

1.2.3. Ad monetization

Rebuilders will monetize with in-app purchases and ads. This opens up another usually unused potential. The most prominent and strongest monetizing form of ads is incentivized ads which are an opt-in form of advertisement. Players can watch ads in order to gain some sort of advantage otherwise granted to paying players. After a certain saturation, there is a diminishing return for the developer as ad networks decrease payment for a player that already watched the same ad before. We can use this capitalistic mechanism and design our reward systems in a way that we can then offer free ad space to organizations that also share our goals. Ideally, we would only show useful ads for free but as with in-app purchase mechanics we need to generate revenue somewhere to even attract potential investors or publishers as none of our team members can work for free.

We're always aware of the tightrope we're walking between exploitative business tactics and positive impact. So sometimes we try to implement this ambivalence in the game to show it. In the case of ads, the character showing ads to the players is called the Propagandapanda.

1.2.4. Revenue

Building on the previously mentioned revenue dilemma: This can be another vector to create a positive impact. Our plan is to actually use our profits for the same goals we have with the game itself. Depending on our funding situation we even played with the thought of turning our company into a non-profit organization as this is pretty much what we will do with our profits anyway. As - at the time of writing - we're still in the development phase and will most probably need investors, the chances for that feasibly happening officially may be slim.

The idea behind using the revenue to support the fight against climate change and pretty much any of the 17 United Nations Sustainable Development Goals (UN SDGs) or strategies set by an organization called "effective altruism" is so obvious to us that it frustrates us every day that other companies with their shareholder structures can't or won't do this. What's the purpose of profit - in a world that is facing so many problems - if it's not tackling them with it?

"Yes, the planet got destroyed. But for a beautiful moment in time we created a lot of value for shareholders."

Figure 2 [Tom Toro, 2015]

There are many ideas that we want to implement with the profits from Rebuilders. They span from lobbying over donations to giving out prizes/grants to players with ideas that advance the fight for a brighter future. We intend to be open with revenue streams and let people see how much revenue we generate, how we do it, and what we use it for.

1.2.5. Data

A game like Rebuilders offers ubiquitous data to potentially pull knowledge from. From testing how to best get players to have positive behavioral changes to testing interface designs or general behavioral analysis.

The idea in this vector is to open up our data for anyone who may need them for scientific purposes. However, as our internal policy is to maximize privacy we will try as much as possible not to collect and definitely not share data that can be traced back to individual users. If this data and our experience can somehow help other studios or research entities to better understand how to make people care about important global problems we'll see this as another win.

2. INTRODUCTION

2.1 MOTIVATION

Developing games for mobile interfaces poses interesting challenges for interaction designs. Desktops, Consoles, and similar devices offer a multitude of different, parallel inputs.

2.2 PROBLEM STATEMENT

Developing games for mobile interfaces poses interesting challenges for interaction designs. Desktops, Consoles, and similar devices offer a multitude of different, parallel inputs. On mobile platforms with touch screens, the maximum number is usually the number of the users‘ fingers. This leads to the next challenge of the fingers concealing parts of the screen. Pointer precision due to finger width and designs of the screens is decreased compared to (for instance) mouses. Such limitations are of special concern for complex interfaces required for 3D modeling.

The case study app „Rebuilders“ features a constructor that players can use to create their own vehicles in a 3D space on a touch interface-driven device. Creating an interaction design for unconstrained touch-based 3D modeling intuitively enough to be instantly proficient is difficult at best and impossible at worst. An example for this would be shapr3D, where, according to email correspondence with the developers, even users with experience in other solid modelers took 5-15 minutes to become proficient. Users without any experience in 3D modeling took up to an entire day.

There are apps for unconstrained 3D modeling available but with a steep learning curve that is not fitting for mainstream mobile games. The challenge is to find a suitable design that combines the freedom of creativity with the right amount of constraints to offer an easy, intuitive interaction.

Designing touch screen interaction without buttons also poses challenges to sketching interface ideas. Scribbling a button or a mouse click is straightforward and understandable, complex multi-touch & drag interactions not so much.

An additional challenge on top of the technical ones is the artwork/visual design of the interface. The style of the interface must adhere to certain aesthetic guidelines and fit the style of the game. Since the style of Rebuilders, in industry terms, is pretty „casual“. It is a low-poly texture-free comic-like look. Therefore it‘s not possible to use a „core“ technical style that might be easier to implement.

2.3 AIM OF THE WORK

The goal is to design a constrained 3D modeling approach that offers enough creative freedom but is simple enough to use to be incorporated into a mobile game for a very broad target audience without any prior experience in designing in a 3D space.
Ease of use is the most important factor because Rebuilders is going to be commercially available with the goal to be profitable and create social impact. Losing players due to bad interface design is not an option.

2.4 APPROACH

The first step is an analysis of existing approaches to get an understanding of what might work and what won‘t.

In a second step, I will try to find general possible approaches and common touch screen interactions to see what gestures or generally what interaction methods (example: finger-swipe) can be used and are common to achieve certain steps (example: object translation).

Thirdly I will design and implement a prototype solution that fits all the criteria for Rebuilders. I prototyped everything immediately at an implementation level. As unity offers such a fast iteration time with its options for integrating assets and testing them quickly, any other forms like paper prototyping wouldn’t have saved much time and would have been impossible due to the required 3D feedback needed.

The last step will be a user study to test the results.

3. METHODOLOGY

3.1 GENERAL

A prototype interface for the constructor of the game „Rebuilders“ will be implemented with unity3D/C# and assets modeled with blender. The focus has to be on simplicity and intuitive handling in order to not shy away potential players.

Evaluation will be achieved by user testings, comparing different interaction approaches, as well as setting up comparisons between desktop and touch screen systems to keep features and limitations in mind during development.

In order to test interface designs where preconceptions of the developer may not lead to the best results, users will be filmed while testing the approach without interference. The goal is to gain qualitative analysis as well as quantitative data.

Users will be asked to complete the tutorial of the game and build a vehicle with a given interaction design while being recorded during and interviewed after testing.
The setup will include a face cam and a camera behind the testers to film their interaction with the mobile device. This way, the data can suggest where players have problems proceeding or where problems arise in the interaction design. Since new implementations will only be iterative, users are expected to be more proficient in each consecutive trial. To mitigate this bias, I will also invite new testers for each consecutive version update in order to see how old testers react to changes as well as new testers that have not yet played around with the interface.

Since there is no work to be built upon for the development of the constructor, the decision on which interaction implementations need to be tested will be based on grounds of expertise as well as quick tests with peers.

3.2 ENVIRONMENT/LANGUAGES

The prototype will be implemented in Unity with C#.
3D Assets will be generated in blender.
2D Assets mostly in Illustrator.

3.3 DESIGN METHODS

For this project, I stuck to a classic approach to UI/UX design. Since there was no prior work to build upon, I opted for the following route:

3.3.1. Requirements gathering

The most impactful part was setting up the general goals of the interface. It should offer extreme flexibility (compared to existing customisability in mobile games) yet ideally be simple enough to not even need a tutorial. Players should feel proficient in the use of the workshop in the span of minutes if not within seconds of using it.

Researching other approaches as mentioned in the analysis of the state of the art led to the conclusion to constrain the design. This has the advantage of still offering very high flexibility in design but decreases the time to mastery significantly. Sadly I can only offer limited data to support this claim in addition to it seeming self-evident.

One example was an iPad App, Shapr3D, that offers unconstrained 3D modeling developed by a team in Budapest. Having contacted them via email I was told their approach takes inexperienced users 1-24 hours to reach a level of proficiency where they could „actually make something they want“. Such a time frame was and is out of the question for my goals.

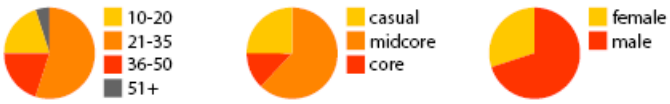
3.3.2. Task analysis

The task, while being very open - „build what you want“, was specific enough to be tested on users once an interface was implemented. Users feeling proficient in placing objects where they want them to be or reconstructing a given example was a sufficient description for my purposes.

3.3.3. User analysis for business case

A brief analysis was done on the future target audience of Rebuilders by comparing the gameplay and style elements with similar games on the market. This analysis is primarily geared towards our business case but also informs our decisions on interface and interaction design.

Analyzing the target audience made things more interesting for interface considerations because the range practically includes all ages proficient enough with smartphones to install and use a mobile game. The one exception would be people not interested in games anyway that may have never experienced the basic hand-eye coordination required for games.
The target audience of Rebuilders will most probably be primarily male - roughly 70% due to the existing demographic in this market segment. Clash of Clans for instance has 77% male players. It does not feel like a big problem for the goals of Rebuilders since according to multiple articles, women already care more about climate change than men do. The differences are not in the same range as 70:30 but on one hand, the 70:30 split is not set in stone and on the other hand, the current game design in a way also reflects our preferences/ skills which skew towards a more mid-core audience.
The target audience also includes all different ages presumably in roughly the following distribution in addition to genre and gender:



3.3.4. Prototyping

After analyzing the state of the art of different constrained 3D modeling approaches my acquired experience as a UI/UX designer was used to start designing solutions for translation and rotation.

3.3.5. Usability inspection

Usability inspection is intertwined with Prototyping(4) because small details were often iterated on before even showing them to a tester. Testing an approach after implementation often already yields results regarding usability, therefore going back to prototyping. After I felt sufficiently comfortable in a design, I let players test it.

3.3.6. User tests

User tests were approached in two different ways.

On one hand, a hand full of people tested the prototypes immediately without data collection besides watching them, taking notes, and discussions.

On the other hand, after many iterations, a more structured test was conducted which is discussed later.

3.3.7. Graphical design

This point is again intertwined with user tests since preliminary unstructured tests were done without concentration on GUI design while the more structured tests were done after GUI design was polished to get feedback on a more final product.
The art style was created out of three requirements:

- look nice
- be cheap to develop
- be performance optimized

After a couple of iterations, this lead to the current art style of Rebuilders - a low poly, flat-shaded approach without the need for textures.

4. STATE OF THE ART

4.1 SUMMARY AND EVALUATION

Games on touch screen platforms follow a multitude of approaches in their interaction designs. Some try to mimic conventional inputs like joysticks, others concentrate on simple clicks. Other games offer fully customized vehicles but so far mostly two-dimensional. Orientation, translation, scale, and rotation are entirely different in 3D space and require new methods. There is research being done on 3D interaction, including porting interface designs to touch devices or sketch-based approaches.

I want to go over a couple of approaches with their respective advantages and disadvantages for the use case of Rebuilders. I will use three more or less subjective metrics to evaluate each approach to compare them at the end: art style, ease of use, and freedom all ranked with the numbers 1-5 like school grades (one being excellent, 5 being bad). There will be no exact catalog to go by to decide on the grades. The subjectivity is enough to generate an overview of the approaches to compare them against each other and to the final approach.

Art style simply describes how closely the look of the objects created by each approach matches what is planned for Rebuilders. Rebuilders will be very low-poly with no rounded edges, circles, or other smooth polygons. An approach that consisted of stacking spheres to model an object would therefore qualify as a five.

Ease of use roughly describes how fast it is to learn the modeling approach and how simplified the interface is. More interface elements, as well as longer learning times, reduce the grade in this aspect.

Freedom describes the possibilities of what can be built by the approaches. A normal 3D modeling tool like blender would get a 1. A 3D pen that only drew unconnected lines in 3D space, even though a very creative tool, would qualify for a worse grade in my case as it would be hard to put any structure to it or modify it. So the main grading mechanism, in this case, is roughly described by „does the player feel like they can express themselves freely?“.

4.2 THE APPROACHES

I identified seven interesting approaches in my literature research about 3D interfaces on touch devices which I will subsequently explain and evaluate informally.

- Shapeshop by Schmidt et al., a simple interface where drawn 2D shapes are “blown up” a bit like balloons or extruded in one axis.
- A suggestive interface by Igarashi et al. where users lay down lines on planes and an algorithm suggests modeling steps dependent upon the laid down lines
- Paper3D by Paczkowski which is analogous to Origami
- Teddy by Igarashi et al. makes it easy to sketch round, comic-like shapes by blowing up 2D shapes to form round 3D objects
- TouchSketch by Wu et al. offers a context-driven 3D manipulation widget
- tBox by Cohé et al. is a neat 3D manipulation widget that combines rotation, translation, and scaling into one interface
- 3D Manipulation widgets by Schmidt et al. which use gestures to switch between different modeling operations

4.2.1. Approach 1 - ShapeShop

Schmidt et al. [Schmidt et al. 2015] were looking for a way to implement a sketch-based 3D modeling tool that offers the possibility to create complex, detailed, and solid models. Their approach lets users draw an outline that is converted into a rough 3D shape that can then be manipulated via additive and subtractive methods. Their data is saved in a Blobtree that holds the information about each modeling step. This offers them the possibility to edit steps in retrospect.

They did not go over their methods of how they tested their approach in comparison to similar sketch-based work but mentioned the advantages of their system over others.

There are three different ways to convert the 2D outline into a 3D shape: „Blobby inflation“, „linear sweeps“ and „surfaces of revolution“.

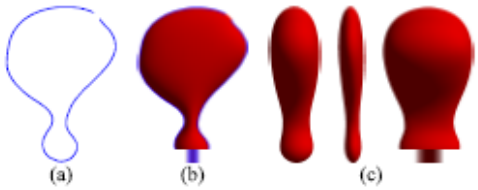


Figure 1: *Blobby inflation converts the 2D sketch shown in (a) into the 3D surface (b) such that the 2D sketch lies on the 3D silhouette. The width of the inflated surface can be manipulated interactively, shown in (c).*

Figure 3 [Schmidt et al. 2015]

From what I could learn, in „Blobby inflation“ the 2D sketch is extruded in a way where the 3D-depth at the line is 0 and increases away from the line until a maximum threshold. This threshold can then be edited to scale it in the third dimension.

Figure 2: *Sketched 2D curves can also be used to create (a) linear sweeps and (b) surfaces of revolution.*

Figure 4 [Schmidt et al. 2015]

Using „linear sweeps“ simply extrudes the 2D surface along the 3D axis for a certain value. „Surfaces of revolution“ applies a revolution modifier revolving the shape around an axis on the 2D plane.

Figure 5

Figure 3: *Cutting can be performed (b) across the object silhoutte or (c) through the object interior. Holes can be interactively translated and rotated. Intersection with other holes is automatically handled, as shown in (d).*

Figure 6 [Schmidt et al. 2015]

Subtractions can be performed by drawing shapes over a 3D body that cuts out an extruded version of the drawn shape.

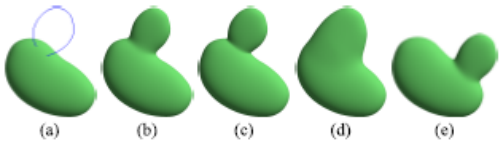


Figure 4: *The sketch-based blending operation (a) creates a new blobby inflation primitive (b) and blends it to the current volume. The blending strength is parameterized and can be interactively manipulated, the extreme settings are shown in (c) and (d). The blend region is recomputed automatically when the blended primitives move, as shown in (e).*

Figure 7 [Schmidt et al. 2015]

Using the same modifier to create the 3D shape, other shapes can be added to it.
This sketch-based system feels like an approach that is relatively easy to learn for players but did not fit at all what is visually envisioned for Rebuilders.

Art style

As this approach can only feasibly generate round shapes with the exception of cut-out edges, I graded the art style with a 4. I could think of an algorithm that approximates user inputs into rectangles but that would make the interaction less intuitive for this approach.

Ease of Use

As the only needed input is a line you draw with your finger I like the simplicity and ease of use which I graded with a 1.

Freedom

The creative freedom in this approach is only missing sharp corners and a bit of control.
I would still grade it with a 2 as it would be a fun tool in a game with enough degrees of freedom.

4.2.2. Approach 2 - A suggestive Interface for 3D Drawing

Igarashi and Hughes designed a suggestive solution to 3D modeling [Igarashi et al. 2001]. Their motivation was to expand on other sketch-based approaches that lack the geometric detail or complexity to design structures like French châteaus. I would call their approach a hybrid between sketch and widget. You can lay out lines and select how to draw the polygons between them or how to extrude them via suggested steps. The software intelligently suggests possibilities to select from depending on how the lines were laid out.

You always start with a line on the floor plane which prompts the software to offer additional steps. Either keep drawing another line on the floor plane or on a plane perpendicular to the previous line. Currently „Château“ supports straight lines and planar polygons only.

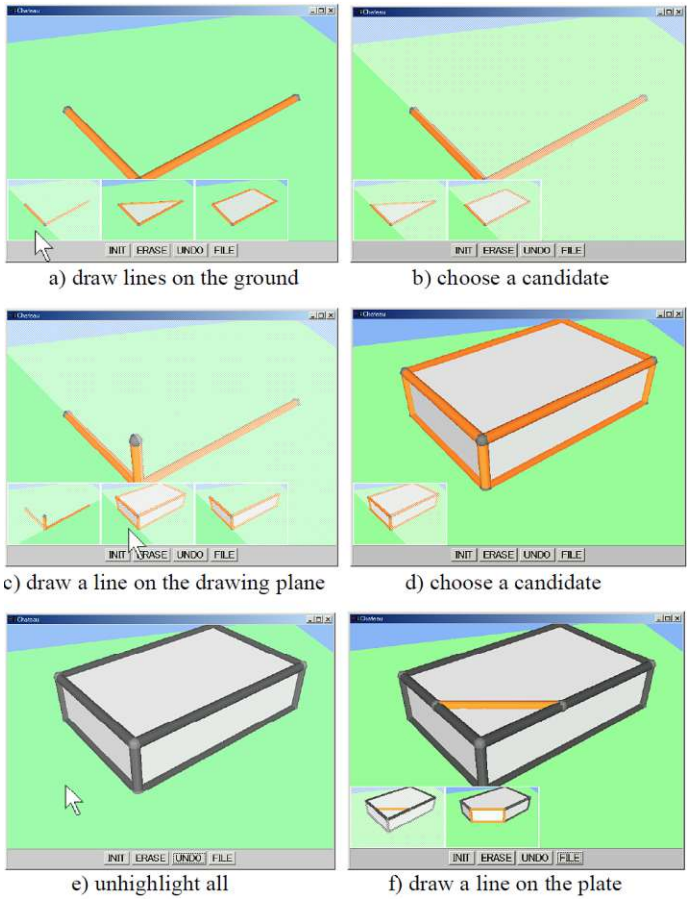


Figure 8 [Igarashi et al. 2001]

I see two parts in this approach that I liked for Rebuilders. The geometric look and the relative ease of the interface. Generally, you only use the right mouse button to draw lines and select suggestions and the left mouse button to navigate the scene. This could have been easily transferred to a touch screen device.

What I did not like is that an operation can lead to a multitude of suggestions and therefore feels a bit „unknown“. Another aspect was an artistic problem. I did not see a nice option of visualizing the lines in the style I envisioned for Rebuilders. Even though in the end I decided against using such an approach I might try a similar approach in a future project.

Art style

The rough and rectangular style fits what I envisioned for Rebuilders but as mentioned, the interface visualizations needed to operate the modeling do not fit it that well. I would still grade it at 2. There may be ways to design the interface in a reduced fashion.

Ease of Use

Even though the primary interaction with drawing lines and selecting suggestions seems easily understandable I see troubles with edge cases and the general approach of building a structure up line by line. Mostly due to the trouble of edge cases where a user may want to achieve a certain goal that is not captured by the suggestions I grade it with a 3.

Freedom

A finished product can theoretically be very complex and there is a lot of creative freedom. As only hard corners are supported I grade it with a 2.

4.2.3. Approach 3 - Paper3D

In Paper3D, Paczkowski et al. also had the goal to create a way to 3D model on multi-touch screens. They focused on tablets and wanted it to be fairly intuitive and casual in comparison to software like Blender or Maya. Their approach [Paczkowski et al. 2014] uses a multitude of interactions to manipulate a two-dimensional plane in 3D space. You fold, cut, and extrude a plane, a bit like cheating in Origami which they based their idea on.

They evaluated their design with four users in comparison to their previously used modeling systems. The overall experience yielded favorable results. They claim that users found it easy to learn, precise enough for casual use cases but not precise enough for professional ones. Mobility (due to the system being on a tablet) was also mentioned as one of the advantages.

Another interesting part of their conclusion is the use of „sub-modes“ which I called context switches in my approach. In Paper3D radio buttons were used to indicate and control which sub-modes were active. Their users did not mind having many sub-modes but Paczkowski et al. also felt the need to optimize the user experience to reduce user interface cluttering which is one of the main goals in Rebuilders as well.

They used six different touch gestures:

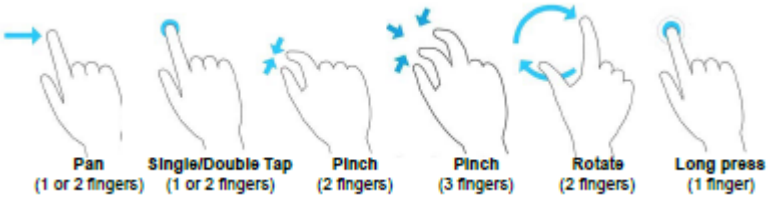


Figure 9 [Paczkowski et al. 2014]

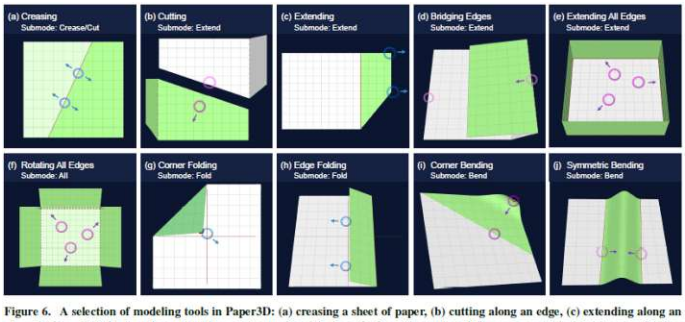


Figure 6. A selection of modeling tools in Paper3D: (a) creasing a sheet of paper, (b) cutting along an edge, (c) extending along an edge of a face, (d) bridging edges of two faces, (e) extending all edges of a face, (f) rotating all faces generated in (e), (g) corner folding, (h) edge folding, (i) corner bending, and (j) symmetric bending. Circles and arrows indicate the position and motion of user touches.

Figure 10 [Paczkowski et al. 2014]

One big issue with this approach is the reliance on relatively big screens due to the more or less finicky selection in the 3D space. As Rebuilders needs to work perfectly well on smaller smartphones too, I will try to remove as many multi-finger gestures and precise multi-touch controls as possible.

Art style

The origami-style is not what I envision for Rebuilders even though the examples in the paper offer quite a range of looks. I grade it with a 4.

Ease of Use

There is a lot to learn here and many different mechanics. The results can be complex and detailed but for a game like Rebuilders I am looking for something far easier. Grade: 4.

Freedom

As this approach pretty much lets you create any shapes at least approximated by an origami-style approach - also with rounded edges - I‘ll rate the freedom with a 1.

4.2.4. Approach 4 - Teddy

Teddy by Igarashi et al. is another sketch-based approach that was referenced by multiple other papers [Igarashi et al. 1999]. Similar to sketch-based approach 1 „shapeshop“ you create a 2D outline that is then „blown“ up. It does offer interactions to remove and add parts to the drawing. Their idea was to create a simple design to generate 3D objects that only need round elements like stuffed animals and they seemingly succeeded as their informal user tests showed that users mastered the operations within 10 minutes and could construct „interesting“ 3D models within minutes.

Figure 11 [Igarashi et al. 1999]

Figure1: Teddy in use on a display-integrated tablet.

Figure 12 [Igarashi et al. 1999]

The difference to approach 1 is its reliance on round shapes where as shapeshop also supported angled shapes. Even though I really liked the simplicity and style of Teddy, it was not what was intended for Rebuilders but it is considered for future projects.

Art style

Even though similar to approach number 1 where you can create round shapes by having simple line inputs lead to 3D geometry, the resulting style is completely round. While I like the simplicity and the look, it is too far from what I envision for Rebuilders and therefore rate it at a 5. As with the technique, I also think about using a similar art style in another project in the future.

Ease of Use

This approach also gets a 1 for ease of use as the only input would be drawing with your finger.

Freedom

While the art style is not what I envision, the creative freedom is still similar or what I would expect a game to offer to players. Therefore I rate it with a 2.

4.2.5. Approach 5 - TouchSketch

As a tool to transform existing objects is also needed because Rebuilders will have pre-built items players can find I had a look at different widget implementations. TouchSketch [Wu et al. 2015] is a simple widget-based approach that offers a 2D interface for one finger to select between axis in order to control a constrained translation, rotation, and scale while the other finger executes the action. Their goal was to offer a 3D manipulation widget for mobile touch screens that offered the same degrees of freedom commonly used widgets in 3D modeling tools. Wu et al. evaluated their approach by a controlled experiment with 12 subjects that had to complete a predefined task in three different implementations while measuring time to completion.

I first envisioned the same evaluation techniques for this thesis but as I progressed through the design it became clearer that time to completion was not my primary focus. Ease of use, as well as time to master, were more important. As I have only found informal approaches to test ease of use I dispensed of the idea to test completion times in a controlled experiment.

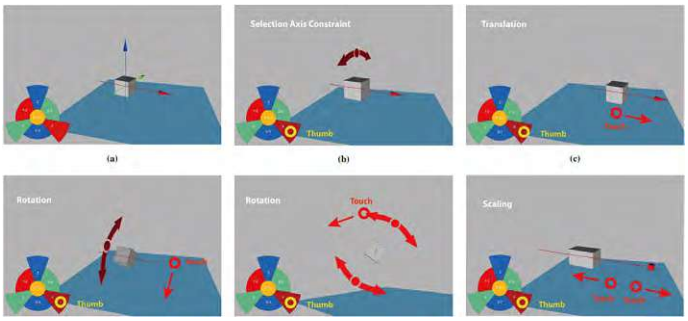


Figure 13 [Wu et al. 2015]

Selecting faces of the object and using the same interactions to manipulate them, the object can be modeled at will.

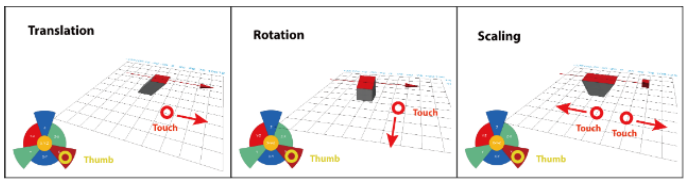


Figure 14 [Wu et al. 2015]

Even though their evaluation showed promising results in comparison to different widget approaches, at this point it did not seem like Rebuilders needed the same degrees of freedom to still offer high levels of creativity. It most probably couldn‘t even offer it due to pre-built assets that were part of the game design. I discarded a reproduction of this approach due to its need for a fairly big 2D widget and its level of complexity that was no longer needed for Rebuilders.

Art style

This interaction method focuses on face manipulation of cubes, the art style would be similar to what I see for Rebuilders. The objects themselves would fit the game even though the interface would need work to fit a smaller mobile touch screen. I still grant a 1 here.

Ease of Use

The different methods of modeling use a variety of buttons and interactions that need a learning process. //TODO: irgendein Problem genauer beschreiben.
Therefore I rate it with a 4.

Freedom

Theoretically, you can create any shapes if there is an extrusion mechanism that can increase the complexity of the asset. For a complex compound shape, the interaction feels a bit too tedious and rounded shapes aren‘t possible. Therefore I rate it with a 3.

4.2.6. Approach 6 - tBox

The closest related work found [Cohé et al. 2011] features touch-based 3D interaction without a context switch or buttons.
Clicking either the mesh itself or the cubical widget, different operations can be applied. Rotation works by starting a drag on the mesh, translation is achieved by starting a drag on a colored axis on the widget and scale by using multi-touch drag on two axes simultaneously.

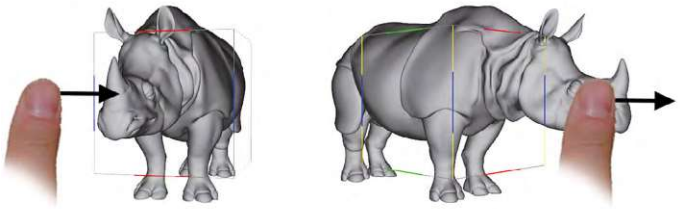


Figure 15 [Cohé et al. 2011]

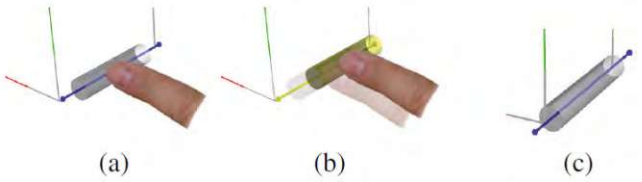


Figure 16 [Cohé et al. 2011]

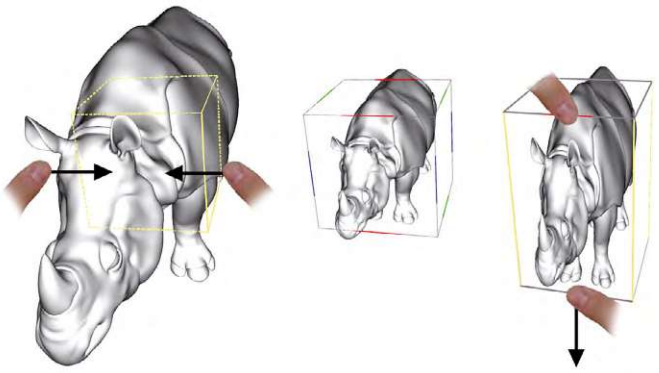


Figure 17 [Cohé et al. 2011]

The need for a dedicated widget to translate the objects still makes it too complex for the design of the constructor. Since translation in my approach will be constrained to pre-existing surfaces (snap to surface), widgets for translation are not needed. As I am trying to reduce as many degrees of freedom in the interaction while still offering a wide range of creative freedom, I will also most probably not use a scaling system.

Art style

The widget itself with its simple lines may be doable in the art style of Rebuilders and combined with the blocky style I envision, it may well work. Since it is dependent upon what objects I use I can only grade the widget itself which I would then do with a 2.

Ease of Use

The interaction is pretty straightforward and easy to learn which I grade with a 2.

Freedom

As this approach strictly speaking is not a modeling approach but a transformation widget it would need to be combined with a modeling approach. Therefore in this case I will just average it out with a 3.

4.2.7. Approach 7 - Sketching and Composing Widgets for 3D Manipulation

Schmidt et al. [Schmidt et al. 2008] tried to create a precise 3D positioning and manipulation widget that worked without buttons. Instead, they used a sketch-based gesture approach that summons the associated widgets. Those widgets are then used to translate/scale/rotate the objects. This is again expected to be too complicated for a mainstream gaming app where users have to be acquainted with the interaction in a matter of seconds.

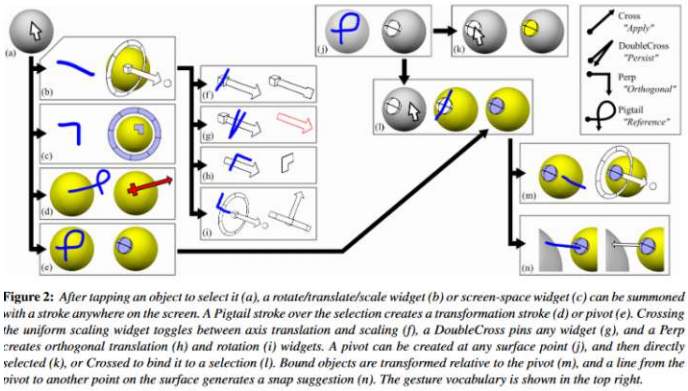


Figure 2: *After tapping an object to select it (a), a rotate/translate/scale widget (b) or screen-space widget (c) can be summoned with a stroke anywhere on the screen. A Pigtail stroke over the selection creates a transformation stroke (d) or pivot (e). Crossing the uniform scaling widget toggles between axis translation and scaling (f), a DoubleCross pins any widget (g), and a Perp creates orthogonal translation (h) and rotation (i) widgets. A pivot can be created at any surface point (j), and then directly selected (k), or Crossed to bind it to a selection (l). Bound objects are transformed relative to the pivot (m), and a line from the pivot to another point on the surface generates a snap suggestion (n). The gesture vocabulary is shown in the top right.*

Figure 18 [Schmidt et al. 2008]

I initially liked the idea of gestures but their conclusions strengthened my opinion that it would be too tough to learn and their widget approach would not be self-explanatory enough for novice users. They tested in three groups of 2 people from experts to novice. The experts were half as fast in a pre-set modeling task compared to conventional 3D software like Maya. The intermediates were just as fast but the novices couldn‘t complete the task as they did not understand the 3D positioning at all. They approached it like a 2D interface and after camera rotation, were wondering why the objects were all over the place as they did not understand the depth concept of the 3D space.

Art style

I can easily imagine implementing a widget like this in the Style of Rebuilders. So it‘s a 1.

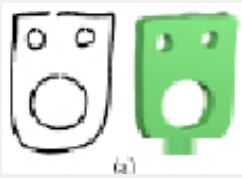
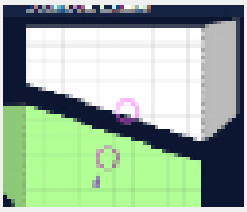
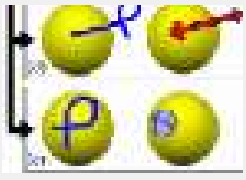
Ease of Use

As novice users couldn‘t figure out how to use it it‘s a clear 5 for me as I need absolute beginners to immediately understand my approach.

Freedom

As with the previous approach, the freedom is dependent upon the actual modeling approach of the meshes and I‘ll simply grade the average again: 3.

4.3 COMPARISON OF THE APPROACHES

approach	art style	ease of use	freedom	average	image reminder
1	4	1	2	2.33	
2	2	3	2	2.33	
3	4	4	1	3	
4	5	1	2	2.33	
5	1	4	3	2.66	
6	2	2	3	2.33	
7	1	5	3	3	

4.3.1. Discussion

Researching all these approaches gave me a better overview of what has been done so far and made it possible for me to compare them to what is envisioned for Rebuilders. This way different interaction methods or UI elements could be ruled out before I tried to implement equivalents myself.

Comparing and grading them gave me a better understanding of the general field as I initially hoped there would be something I could rely on for my work on Rebuilders. This turned out not to be the case as all had their strengths and weaknesses and averaged out >2.5. It would have been easier to find an approach that already filled my needs and iterate on it. Since this general 3D interaction style is still new on touch devices, especially with the requirements of a rather casual game that needs to fit a broad range of users, I needed to come up with something different.

During my work on Rebuilders other games were published that also implemented some approaches towards generating 3D objects / spaces. I will have a brief look at those in the next chapter to see how they approached their interaction design.

5. SIMILAR SOLUTION

In this chapter, I quickly gloss over other games that were found mostly while already working on Rebuilders since when I started with the process there were zero other games with 3D interfaces on the market. I still tried to research how they found solutions to their issues or how they constrained their approaches in order to work with a broad and untrained audience.

5.1 MINECRAFT(-LIKE) MOBILE

When speaking of „Minecraft“ in this section, games are meant that are practically clones of Minecraft mobile. To test the mechanics I resorted to free versions that implemented the same kind of interactions.

There are games that utilize 3D construction mechanics but in a different fashion with fewer degrees of freedom. Minecraft is such an example. There is only one type of block, a cube, that can be positioned. This makes interaction fairly easy. There is no need to introduce a context switch or buttons for different interactions. Not needing to further interact with a specific block also removes the need to have an „active“ element that you interact with. You just remove blocks or add them.

Translation and rotation of the scene work by introducing a character that traverses the 3D space and wherever it looks you can place blocks by tapping your finger on the screen on the desired position. Blocks then appear in that spot.

Figure 19 source: Google Play: „crafting and building“

Holding your finger longer over a block removes them.

Figure 20 source: Google Play: „crafting and building"

There is an advantage to this method: speed. You can quickly place blocks into the scene but with this method, there is no simple way to differentiate between selecting a block or placing one. There would have to be a context switch between selection and placement.
Another difficulty arises when you want to rotate elements, which is needed for Rebuilders. Another context switch would be needed.

While selection does not play a role in Minecraft, Rebuilders needs to offer the players to reposition parts on characters and vehicles. In Minecraft simply removing a block and placing a new one somewhere else is a viable strategy because they are all positioned in a grid, are the same size, and don't need to be placed continuously like items in Rebuilders.

When placing something with more creative freedom, deleting and replacing something you just want to move a bit to one side or rotate by a couple of degrees makes little sense. Especially if it's about rotating it since you'd need to go into rotation mode again anyway. Therefore a selection mechanic is needed.

One of my hypotheses is, that there is a more intuitive form of interaction for my case that does not require as many context switches which will be tested in this project. I cannot however test it against the Minecraft approach since rotation is a requirement and there are different elements to be placed which do not exist in Minecraft.

6. GENERAL INTERACTIONS

As previously mentioned, it is not easy to find research in the area of 3D interaction on mobile devices. They are still rare even at the time of writing.

To support my initial assumptions on how to design the interactions I also looked at existing research in the space of mobile touch screens, which mostly operates in two dimensions.

What the research I found has in common is a list of generally accepted interactions:

- click
- drag
- pinch/expand
- two-finger pan
- two-finger rotate
- gestures
- click&hold

There are interesting findings when comparing the different methods in [Kobayashi et al. 2011] showing that elderly people preferred dragging over clicking. When adding other research like [Hoggan et al. 2013] or [Li et al. 2005] it seems to be possible to create a rough general ranking of possible interactions by difficulty.

interaction	difficulty (rank, low=more difficult)
two-finger rotate	1
click&hold	2
two-finger pan	3
pinch/expand	4
click	5
drag/gestures	6

This way features of the workshop could be prioritised and then interactions applied to them in increasing difficulty, meaning the most important features would be covered by the easiest interactions.

Interactions of Rebuilders were then firstly prioritized roughly by how often you will most probably use each feature. Then finger interactions were added from easy to hard but the easy ones were used as often as possible depending on the interface. To give an example: Positioning an item would be achieved by dragging it with one finger from a shelf to the vehicle. Rotating the camera can also be achieved by a one-finger drag that just starts somewhere else but on the item to be positioned. So both interactions can use the same easy finger movement.

Switching context between rotating and positioning items then uses a one-finger click. The click is also used for different buttons on the interface. Item rotation then uses the drag again as we are in a different context.

The context was important for me as I could not figure out a way to rotate elements without a widget and just by using a different finger input. As the rotation of Blocks needs to be performed on 3 axes it cannot be simply mapped onto a 2-dimensional interface input. Either a widget is created that offers 3 axes (which needs a context switch) or another context switch that is called by a different finger input for each rotation axis. As this would only complicate things I decided on the simple context switch between positioning and rotating.

feature	priority	interaction
item positioning	1	drag
camera rotation	2	drag
context switch	3	click
item rotation	4	drag
camera zoom	5	pinch/expand
camera pan	6	two-finger pan

Interestingly enough the resulting match was no different than the one I would have intuitively created.

This ranking however does not necessarily produce the most intuitive outcome. I will later discuss camera pan and context switch between positioning and rotating in more detail when discussing the results of user tests.

7. PREVIOUS APPROACHES

Before getting to the suggested solution I also want to show the tried and tested steps that were more or less quickly discarded for one or more reasons to iterate on the development steps. I will describe the ideas behind the tried solutions, how they worked, and give Pros and Cons that came from a mix of tests with peers as well as users. The approach in each design step I will outline below started with a new idea that I first implemented in Unity, tested myself and once it was bug-free I gave it to my peers without giving any information and watching them play with it. If the feedback was conflicting I looked for a handful of players in my vicinity and let them informally test it the way my peers did. Once I had sufficient feedback - in the best case, negative feedback I could learn from - I either went on to a different approach, kept parts of the old one, or stuck to it.

7.1 BLOCK ROTATION

Rotating Blocks was the biggest challenge. That‘s why I dedicated time to test 5 different implementations lined out below.

7.1.1. Arrows to click

The first iteration consisted of flat arrows that were placed on the block‘s faces. Four on each face that pointed outwards towards the edges. Clicking one of those arrows instantly rotated the block 90 degrees in that direction.

This approach lead to a pivotal design decision for subsequent approaches. As I will discuss in the Cons of this design, instantly rotating objects in 90-degree increments without showing a continuous animation of the rotation proved to be a bad idea.

Pros:
Initiating the rotation worked pretty intuitively. Players immediately tried to click the arrows which rotated the blocks.

Cons:
Following the block‘s rotation without having a smooth continuous transition was surprisingly difficult. It seems to require a user to have fairly good spatial awareness to keep track of instant rotations without the visual cues a continuous rotation offers. This was most probably due to the primary objects being cube-shaped without many distinct features. I assume instantly rotating a more complex mesh (i.e.. a human body) would work fairly well. But as even I had my troubles rotating in this way, additionally to the user feedback, I immediately discarded this approach in further designs.

Conclusion:
A continuous rotation is needed. Clicking a rotation button would mean I’d need a timed animation that introduces new challenges. I discarded the idea of clicking a button and also went with a drag&drop mechanic. That way the animation follows the player’s finger.

7.1.2. No widget

As I decided to get rid of discrete rotation, a click would not work anymore. So the first new idea was to have continuous rotation without any widget. Here I did not introduce any interface element. When the user starts to drag a block when in rotation mode, the direction in which the finger is dragged on the screen was correlated to the 3D plane of the face where the drag started. The corresponding direction within a 90-degree quadrant then led the block to be rotated on the corresponding axis.

This was an attempt to intuitively rotate blocks without any further interface elements. In essence: you start to drag a block and its rotation follows your finger.

Pros:
The rotation worked pretty intuitively besides a couple of edge cases.
The interaction felt more „direct“ because you interacted with the blocks directly.

Cons:
Players just didn‘t know what to do without a visual cue. A pretty obvious oversight on my part. It would have needed a strong UI cue for players to know that they are now in rotation mode.

Conclusion:
Interface elements are needed.

7.1.3. Arrows appear once dragged

After seeing that an interface element should be introduced I tried to combine the no-widget approach with a visual cue of what was going on after starting to drag. Even though the initial cue was still missing I wanted to see whether an interface element once dragged would speed up the learning curve or make it clearer what was happening. When you started to drag and rotate a block, an arrow appeared on the dragged face that pointed in the rotation direction.

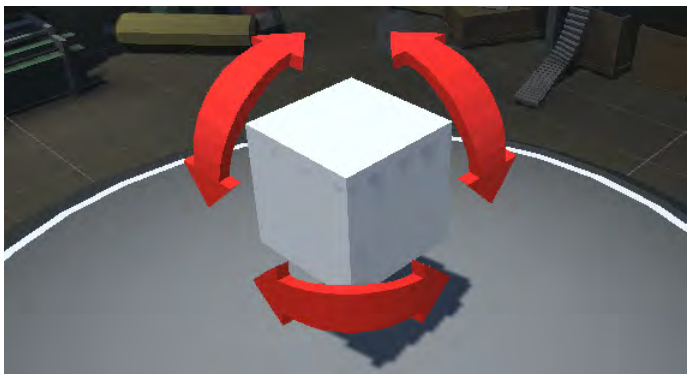
Pros:
It worked intuitively for users that knew what to do.

Cons:
This unsurprisingly did not fix the problems of the „no widget“ approach since the problem turned out to be the missing initial visual cue, not a visualization of what was happening while rotating a block. Users that did not know what to do still didn‘t figure out how to rotate a block.

Conclusions:
I tried again. But as before.. interface elements needed.

7.1.4. Arrows over edges

In this design, I tried putting rounded arrows in 3D space over block edges facing in both rotation directions. In essence, the player picked a rotation axis by starting to drag on one of those two directional arrows over an edge. This was there was continuous rotation and a widget to show the players what they could do.



Depending on the camera angle all other arrows were hidden so only the three necessary for the rotation were visible. I tried two different approaches here:

Arrows over edges - static arrows

The UI element stayed once a drag was started on an arrow while the block rotated.

Arrows over edges - dynamic arrow

Once a drag was started, the other two arrows vanished and the one dragged arrow would move with the block to further indicate which axis was selected.

Pros:
Initiating the rotation worked fairly well and the brain could easily keep up with the rotation.

Cons:
It felt less direct using this sort of widget.
The arrows occupied a bit more visual space than I would have liked for the design.
When working at a camera angle perpendicular to a face, a small change in the angle to a different one could force a switch between the arrows. This was less of a usability issue than it was a visual nuisance.

Conclusion:
Both dynamic and static arrows worked fairly well. I liked the style but as there were a couple of small Cons, I wanted to keep testing other approaches.

7.1.5. 3D circular widget

To test a more familiar approach I briefly tried to introduce a 3D widget like in Maya, Blender, or similar 3D modeling software where 3 differently colored circles marked the corresponding rotation axis. I discarded this approach mostly due to its lack of visual appeal.

Pros:
Users that were familiar or at least have seen 3D modeling software immediately knew what to do.

Cons:
Rotating a block worked mediocrely. Having the camera aligned with one of the rotation planes made it hard to use this rotation plane. In other words: When looking at a block from the side you couldn‘t rotate it around the vertical axis well because it was just the width of the circular widget that you could use. I even gave the circles a thickness greater than zero in one update but that just didn‘t fix it.

Conclusion:
The Cons were overwhelming and I also did not like the style and couldn’t figure out a design that worked well. I discarded it.

7.2 ITEM ROTATION

As Rebuilders also offers items that can later be placed on the base consisting of the positioned blocks, interaction methods for them needed to be designed as well. The user experience of Item rotation grew with that of block rotation in the sense that it first had big red arrows to rotate them and with the switch to a small widget and rotation by starting the drag on the block I also implement the rotation in that way too. So by working on block rotation techniques I could transfer the findings over to item rotation techniques. Since items always only have one rotational axis I used the same interface element that shows the current rotation axis while rotating a block to communicate to the users that they are now in rotation mode.

7.3 BLOCK TRANSLATION

The biggest difficulty in positioning blocks came from the problem of having your finger over the block while dragging it where you want it to be, effectively blocking your view.
I implemented an offset between the finger and a block that‘s dragged from the shelf to the vehicle. This immediately led to another issue: When dragging an already existing block to a new position, the offset made the block „jump“ when starting to drag it, which was unexpected and a bit unnerving, especially for items. Therefore I decided to only ad the offset when adding a new block from the shelf to the vehicle but not when repositioning an existing block.

7.4 ITEM TRANSLATION

From an interface standpoint, I designed the translation of items exactly the same way as with blocks. The big difference is that items can be positioned continuously on the surface that was created by positioning blocks while blocks stuck to a grid.

7.5 ELEMENT DELETION

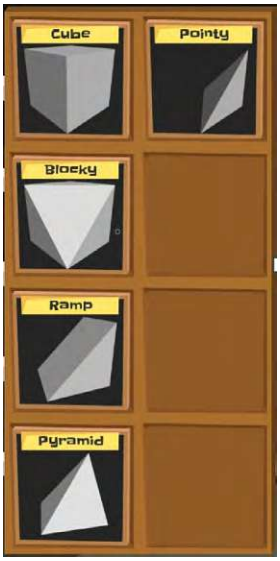
I previously had an interface element in the form of a recycling bin where players dragged blocks or items over in order to delete them. This had the advantage, that when accidentally dragging a block off the vehicle, it jumped back to where it was before. After implementing an undo/redo function and seeing a couple of users try to drag blocks off the vehicle to delete them I decided this seemed to no longer be needed.

8. SUGGESTED SOLUTION

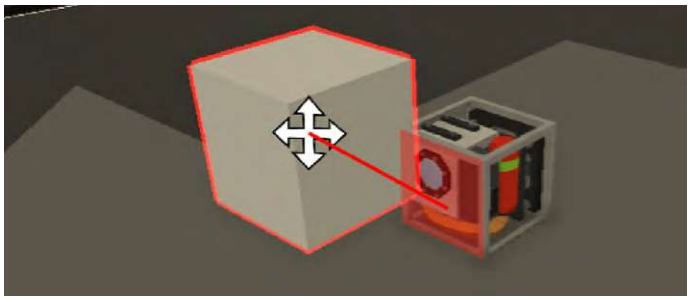
Taking the previous findings as well as the literature research into account I came up with the following proposed implementation. I will describe the different mechanics that I finally implemented and go over User tests in the subsequent chapter.

8.1 PLACEMENT GRID

Using Unity/C# as an engine/code framework plus Blender 3D for the assets I implemented the first proof of concept with five different blocks, each missing one vertex more than the previous one:



The implemented solution always starts with a cube that players use to build upon. When dragging a new block into the scene, a Raycast is used to find the position in the grid for the new block.



If the raycast centered on the block hits a face of the primeblock (or any other block), here highlighted in red, it will snap to that face and occupy the corresponding position in the grid as follows:

I mentioned a „grid“ because as previously discussed, blocks always snap to other blocks with their center points in a perfect quadratic 3D grid. Or in other words: two cubes will always share a complete side with 4 vertices in the same position. Smaller blocks will follow the same rules as if they were a cube.

8.2 BLOCK TRANSLATION

Translating a block works by clicking it once to select it and then dragging it around. This offers the advantage of being able to rotate the scene by beginning to drag from any finger position on the screen without accidentally starting to reposition blocks. Once in translation mode, a translation widget is rendered over the selected cube and it is outlined to visualize that you are in a state with an active block you can now manipulate. Otherwise, the user would not know what to do or expect and could not differentiate between scene rotation and block translation modes.

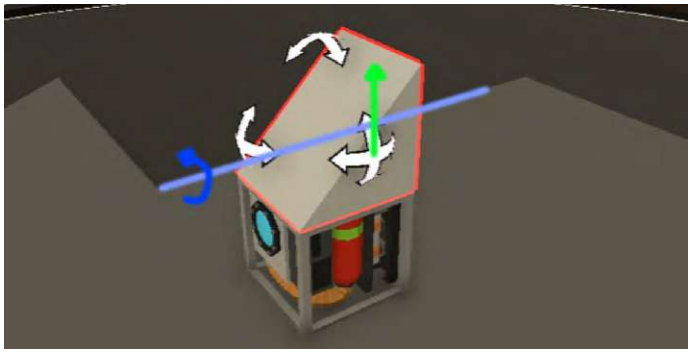
The user can then drag the block by touching it and moving it around. Wherever a raycast hits another block, the current block automatically snaps to the corresponding grid position. When releasing the finger, the block stays where it was at that time.

8.3 BLOCK DELETION

In case a block was not on any grid position while being repositioned (dragged away from the current construct) it is deleted.

8.4 ROTATION

Incorporating the findings of the previous chapter, rotation now works by clicking a block again after selecting it for translation. Clicking it again switches back to translation.
When in rotation mode, widgets appear that indicate that you can now rotate a block. After a few iterations, I set for a simple white rotation “cross“ along the faces to indicate different rotation options.



(green: finger movement, blue: rotation axis)

When starting to drag on a face, there are only two valid rotation axis. The rotation axis perpendicular to the face wouldn‘t make much sense because a user expects to rotate a block by dragging a finger across the screen and have the object follow in a „haptic“ manner. In order to achieve this, when starting the drag, the direction of the finger movement is used to

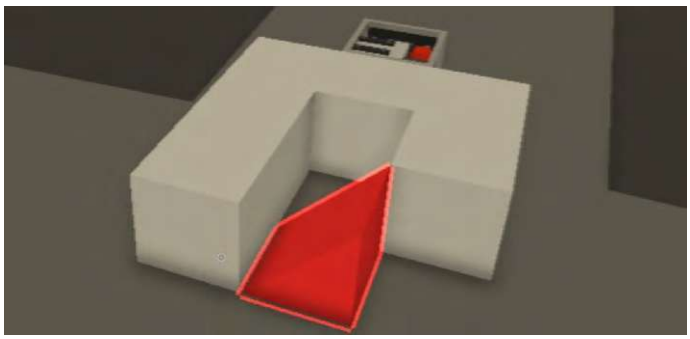
define the rotation axis in accordance with the widget. When releasing the finger, the block snaps to the closest 90° increment.

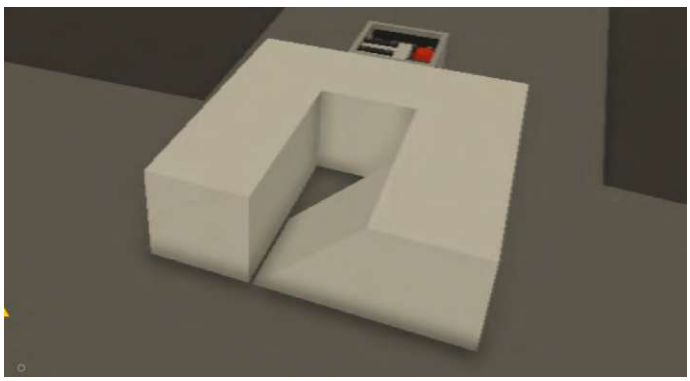
8.5 SCALING

After going over existing approaches and iterating on the art style of Rebuilders I came to the conclusion that scaling objects would not be required to offer the wanted freedom in design. The advantage is lowering the complexity of the interface even more while also supporting a more coherent outcome of the created characters and vehicles. Or in simpler words: It gives me more control over the art style.

8.6 VALIDITY CHECK

For the use case of Rebuilders, blocks need to have overlapping surfaces big enough to theoretically be 3D printed in the future. Therefore a connection as in the first image is only valid if it is connected via a route like in the second picture:





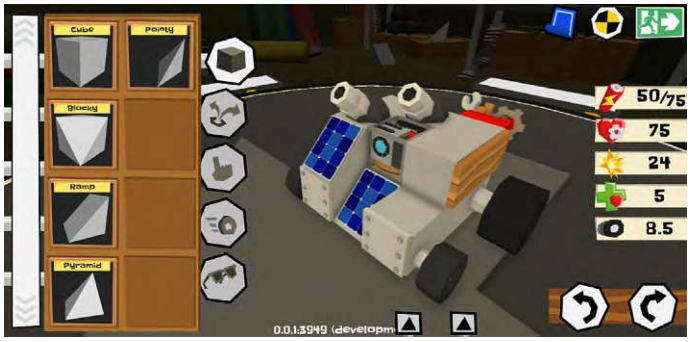
I wrote a recursive algorithm going through the blocks, counting their connected faces, and memorizing the route. That way even when two Blocks had a connection that was not legitimate, they could become marked as legitimate when traversed through another connection like in the images above.

8.7 ITEM POSITIONING

Placing items works similarly with more degrees of freedom in translation, only one rotation axis but this one with a continuous rotation and not in 90-degree increments like with blocks. Players can position items anywhere along the surface of a block-mesh:

When rotating them, they all have only one rotation axis perpendicular to the face they‘re snapped to:

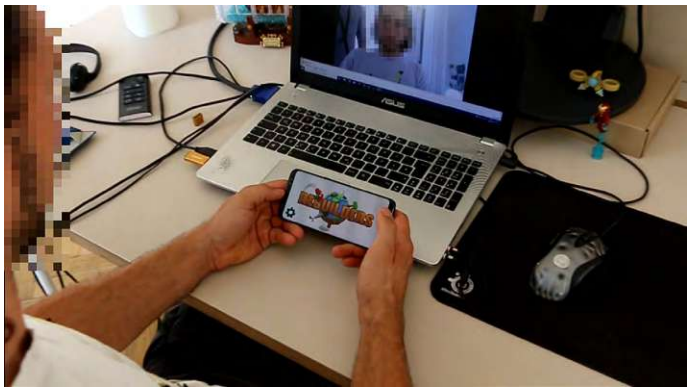
This constrained approach offers an interaction where vehicles can be created freely, easily, and still with a lot of creativity involved.



9. USER TESTS

9.1 PHYSICAL SETUP

The setup was comprised of two cameras, one filming the face of the testers and the other an over-the-shoulder view to see the interaction of both hands with the mobile device.
The test device had a screen aspect ratio of 2.05:1 with a diagonal screen size of 147.3 mm.



9.2 STRUCTURAL SETUP

The workshop had a rudimentary tutorial that future players will also receive to show them mode-switches between positioning and rotation. It wasn't expected that players find out such details of the interaction design by themselves even though in hindsight it would have been an interesting test. I will follow up on this in future tests.

After a guided tutorial players had to build their own vehicles from scratch to see if they felt proficient in using the workshop immediately.

I stood behind testers and only interfered or gave pointers when there was either a bug or the user was stuck and asked for guidance. Meanwhile I also took notes (on paper) of issues I could immediately see in case I might miss them during video analysis.

9.3 BASE DATA

My primary goal was to have user tests in order to find previously unknown issues. Once no new issues are found I will keep going for three more players to have a buffer in case some results were inconclusive. Already after 4 user tests no data on new errors/issues could be gathered and I settled for 7 user tests since finding issues and testing the feeling of proficiency was the main goal.

The tester‘s ages were roughly 8, 13, 4x30 and 65.

Three female participants and 4 male participants. This is an over-representation of female players in the anticipated target audience but results showed no noticeable difference in issues/proficiency between these two sexes. The initial proficiencies varied widely. Three of the testers never play mobile games, two frequently and the other two very infrequently. From all the testers only one had experience with games that offered a similar tinkering experience.

Each subject was assigned a subjective value of 1-5 (1 = highly proficient, 5 = never really played digital games) in perceived proficiency of comparable games just to see if there was a difference in scores between those levels.

subject	age	gender	tutorial completion time	experience lvl	issue 1 (context)	issue 2 (top-down)	issue 3 (camera pan)	issue 4 (raycast)	issue 5 (wheels)
s1	8	w	01:37	4	X	X	X	X	
s2	30	w	01:10	3					X
s3	30	w	01:02	2	X			X	
s4	13	m	01:05	1		X			X
s5	30	m	03:04	2			X		X
s6	30	m	03:08	4	X	X		X	X
s7	65	m	01:30	4			X	X	X

9.4 ANALYSIS

First some general time data was captured to compare users and also to see if there was an unexpectedly high variance.

Going through the notes and videos I tried to find issues in the interface design common to the testers and see if and how they reacted to them.

The two main aspects I hoped to find in addition to issues that I could correct, were acceptable times until proficiency as well as a positive subjective feeling of proficiency.

9.5 RESULTS

The median time to finish the tutorial was one minute and twenty seconds. The mean time was higher (01:49) due to encountered interface bugs of two users. Excluding those two, there was no substantial difference in completion times.

After analyzing the video material and checking for common issues with all testers, 5 major issues came up that have to be addressed.

I will go over the issues by first describing what was observed, give a brief overview of how different subjects reacted to it, then analyze what happened and try to offer a solution that would fix the issue.

9.5.1. Issue 1: context switch click

Observation

As previously described, to rotate an item or a block a player has to switch between these two modes by clicking the object. When in positioning mode, another click switches to rotation and vice versa. Even though shown in the tutorial, testers forgot how to rotate items after the tutorial.

subject	interaction
s1	s1 seemed confused when she first tried to rotate an object again after the tutorial, but after clicking around, she figured it out again
s2	
s3	s3 sometimes seemed to have a hard time switching context between rotating and scaling, may have to differentiate stronger optically?
s4	
s5	
s6	s6 also did not catch the rotate mechanics, tried to rotate it by dragging while in position mode once, then figured out the click again
s7	

Analysis

I found two reasons for this:
The tutorial showed how to select an object and then explained how to manipulate its position. After this, it was shown to click the object to rotate it. Apparently not mentioning the context switch specifically and explaining, that you have to click *again*, it was easy to miss the connection for the context. The brain seems to just memorize „click to select“ and „click to rotate“ in this scenario but not make the connection „click to rotate when already in positioning mode“. Some players were confused when they tried to rotate again, others clicked again but wildly and jumped between modes.

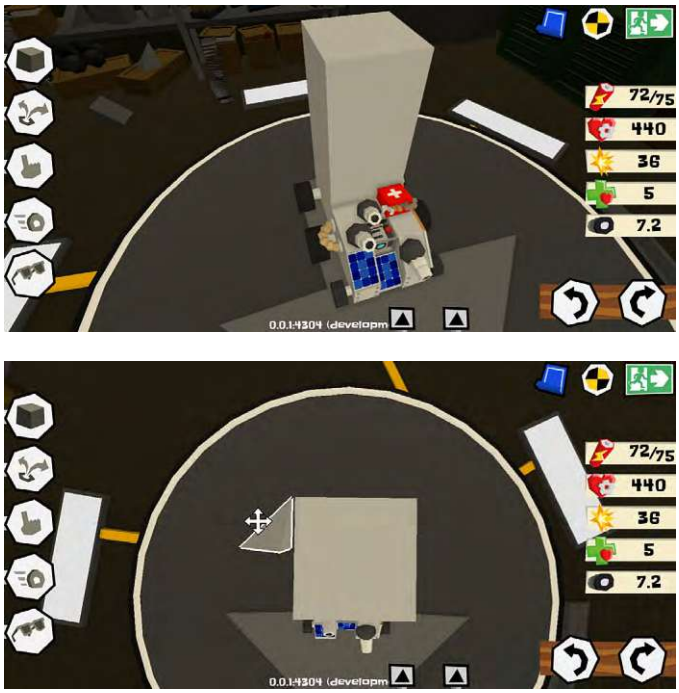
Solution:

The first test in the future will be to have a different sequence of events in the tutorial and let the player rotate an object before positioning it. This way they have to click through the positioning context and then back again which may increase retention of this feature.
Another option would be to additionally reintroduce an interface element that was previously there to switch between the two contexts. It was a rendition of a pedestrian traffic light that offered two buttons, one for positioning, one for rotation. I opted for its removal because I imagined players to be faster clicking on the item itself instead of navigating to an interface element. Having both solutions may deal with this problem but will clutter the interface.

9.5.2. Issue 2: positioning top-down

Observation

Since humans obviously aren‘t accustomed to raycast tests, another issue arose that was not anticipated. When rotating the camera in a top-down position, meaning the camera axis is aligned with the Y-axis, players tried to position a block on the edge of a vehicle where there was no block below it to raycast onto.



subject	interaction
s1	s1 rotated the camera to the top side and tried to position a block on the edge of the vehicle and expected a snap
s2	
s3	
s4	s4 did the same thing. The error was always the same, as was the reaction to it.
s5	
s6	same as s4
s7	

Analysis

Players expected the software to snap the block to a position it could not know. Since the vehicle can have multiple layers of blocks, the algorithm cannot guess which height which layer the block should snap to.

Usually, the user wanted it to snap to the top-most layer.
Interestingly this problem mainly happened in the top-down perspective. When looking at the vehicle from the side, the raycast seemed to be more intuitive for players. The sample size was too small to infer whether this was a common phenomenon. And since a solution wasn‘t dependent on finding it out, I decided not to follow up on it yet.

Solution

A proximity snap function will be implemented where blocks snap to fitting positions in the grid when a certain threshold is reached. This way, when looking from above and the block is almost where the player wants it to be it will snap without a raycast hit. One issue may remain: The position of the block while „holding it in your finger“ is an arbitrary distance to the camera that was selected by me. So players without a good enough spatial awareness may still expect the block to snap to the edge even though there is still space between them along the view-axis.

The only further solution if such a problem arises would be to always snap them to the top-most level of the vehicle‘s edge in this edge case (no pun intended).

9.5.3. Issue 3: camera interaction (pan)

Observation

The camera interaction for panning was not as intuitive as hoped. Three of the seven testers did not figure out that you can pan the camera with a two-finger drag without explanation within the first few minutes.

subject	interaction
s1	never panned the camera and just built with zoom and rotation until it was mentioned to them
s2	
s3	
s4	
s5	only accidentally figured panning out after a while while zooming
s6	
s7	never panned until mentioned

Analysis

Rotation was immediately apparent since it is only one finger dragging across the screen. Zooming by pinching two fingers was easy as well since it is a widespread mobile or touch screen mechanic.

With the camera pan, additional issues were expected going forward (even though they did not happen during tests with players) because while panning you can move the pivot point of where your camera is centered. That way you could „fly“ away with the camera and get to a situation where you cannot work on the vehicle usefully. A bounding box was introduced and when re-entering the workshop the camera resets. This will need further testing.

Solution

A simple solution would be to instruct players how to pan the camera during the tutorial/ onboarding process with a guided step. I currently have no concrete interaction-intrinsic solution in mind for this problem. A swap between rotation and panning will be tested in a later phase. When you use one finger, you pan and since rotation is something every user seems to expect, they might try with two fingers while rotating.

Players would most probably have eventually realized how to pan but even three of the testers not figuring it out - in the time I was hoping them to become proficient - is enough for me to consider testing alternatives.

9.5.4. Issue 4: raycast hit

Observation

Close to issue 2, players sometimes had difficulty positioning blocks on other angled blocks where they expected the block to already snap but didn‘t have a raycast hit behind.

subject	interaction
s1	s1 sometimes had trouble snapping blocks to others from angles where the raycast hit did not actually hit the expected place of the object
s2	
s3	s3 had similar issues. Again, seems to be no different to other testers. It is simply not understood how the snapping works
s4	
s5	
s6	same as the others. needs proximity snapping.
s7	same as the others

Analysis

Players expected the algorithm to position the block into the grid when they were already close to its grid position. Sometimes, depending on the camera angle and the geometry, the raycast would not hit anything in its line of sight while the block was already close to where the player wanted it to be placed.

Solution

Similar to issue 2, there will be a general snap mechanic when a block is in proximity to a grid position. This way, blocks will snap more often but this should not be an issue as they can still be dragged along wherever the player wants them to be snapped to. It will reduce the frustration of not understanding why a block won’t snap to where the player expects it to.

9.5.5. Issue 5: item wheel snapping

Observation

A majority of players complained about wheels not snapping. This was partly expected. Wheels in Rebuilders behave like any other item and can be positioned freely. When players embraced the fact that wheels could be positioned without a physical need for them to be in positions where you would expect them to be in - and placed them comically on top of the vehicle - the need for snapping disappeared. But this need stayed when building them in „normal“ positions. Players did get used to it but it was enough of a nuisance for them to voice their opinion about it.

subject	interaction
s1	
s2	s2 didn't expect the tires to snap but wished for them to snap
s3	
s4	s4 thought the tires would snap first but understood why they didn't but still thought it would be better if they snapped
s5	s5 asked if the position of tires had an effect on the vehicles performance and thought they should snap
s6	s6 same as s5
s7	same as the others

Analysis

The problem is that people expect wheels to be in certain positions. Specifically: on the opposite side of another wheel. When they positioned a wheel on the other side of an already positioned wheel they expected the second wheel to snap in position on the axis of the wheel on the opposite side.

Solution

A soft snapping mechanic will be implemented for wheels when there is another wheel in the opposite direction somewhere in the vicinity of its axis and then snap to the corresponding position. This way wheels can still be positioned almost as creatively as before but when players position them like normal wheels, the result will be a snap when close to an axis of an opposite wheel.

10. CONCLUSIONS AND FUTURE WORK

10.1 CONCLUSIONS

The 3D workshop turned out as hoped. Players feel proficient quickly enough to have it be part of the game. There are still small improvements to be made in the UX design in parts of the interface that were not discussed here but overall players were satisfied with the experience.

I cannot generalize from this conclusion but I still feel that the selected approach/ methodology was very helpful in the overall design process even if contextually limited in this particular use case. The feedback gave me many points to work on in the future around the workshop interface.

In hindsight, I could have implemented different solutions comparable to an A/B testing approach and compared the time to complete a task between them. However, since soon after the first iterations, user tests showed that the approach already worked well it would most probably have taken more time than it would have yielded results that were worthwhile. Ranking a mix between objective (time to completion) and subjective (visual appeal of the interface) would also have been hard to code objectively.

Choosing between a solution that had a good objective/quantitative measurement but a bad subjective/qualitative one and vice versa would again have been a subjective (mine) decision in the end, because a solution that‘s faster but might feel worse doesn‘t have to be the better solution.

It seems like the design process / my approach up until I started the user tests, was already fairly well rounded. I hoped to gain more knowledge during the trials and find more mistakes that I could have worked on. All in all the problems that arose could be fixed or are in the pipeline to be fixed and players felt proficient in their modeling approaches in a very short amount of time.

10.2 FUTURE WORK

After an initial smaller launch we will have plenty of data to work on improvements but I will not pursue these changes in an academic setting. I will most probably not be the person working on the interface in the future due to my other responsibilities.
Though building on the rich data sets we can generate with Rebuilders, there is plenty of other academic potential ahead that will probably pique my interest.

11. APPENDIX

The testversion is publicly available in Austria in the Google Play Store.
https://play.google.com/store/apps/details?id=cc.rebuilders

12. SOURCES

12.1 IN TEXT

- R. Schmidt, B./Wyvill/M. C. Sousa/J. A. Jorge, ShapeShop: Sketch-Based Solid Modeling with BlobTrees, 2015
- Takeo Igarashi/John F. Hughes, A Suggestive Interface for 3D Drawing, 2001
- Patrick Paczkowski/Julie Dorsey/Holly Rushmeier/Min H. Kim, Paper3D: Bringing Casual 3D Modeling to a Multi-Touch Interface, 2014
- Takeo Igarashi/Satoshi Matsuoka/Hidehiko Tanaka, Teddy: A Sketching Interface for 3D Freeform Design, 1999
- Siju Wu/Amine Chellali/Samir Otmane/Guillaume Moreau, TouchSketch: a touch-based interface for 3D object manipulation and editing, 2015
- Cohé A./DECLE F./HACHET M., tbox: A 3D transformation widget designed for touch-screens, 2011
- Ryan Schmidt/Karan Singh/Ravin Balakrishnan, Sketching and Composing Widgets for 3D Manipulation, 2008
- Masatomo Kobayashi/Atsushi Hiyama/Takahiro Miura/Chieko Asakawa/Michitaka Hirose/Tohru Ifukube, Elderly User Evaluation of Mobile Touchscreen Interactions, 2011
- Eve Hoggan/John Williamson/Antti Oulasvirta/Miguel Nacenta/Per Ola Kristensson/Anu Lehtiö, Multi-Touch Rotation Gestures: Performance and Ergonomics, 2013
- Li, Y./Hinckley, K./Guan, Z., Experimental analysis of mode switching techniques in pen-based user interfaces, 2005

12.2 CONTEXT

- Florence Balagtas-Fernandez/Jenny Forrai/Heinrich Hussmann, Evaluation of User Interface Design and Input Methods for Applications on Mobile Touch Screen Devices, 2009
- Lumpapun Punchoojit/Nuttanont Hongwarittorrn, Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review, 2017
- Andrew Bragdon, Eugene Nelson, Yang Li, Ken Hinckley
  Experimental Analysis of Touch Screen Gesture Designs in Mobile Environments
- Keith B. Perry, Juan Pablo Hourcade
  Evaluating One Handed Thumb Tapping on Mobile Touchscreen Devices