CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Designing Synchronised Multi-user Augmented Reality Experiences

A Collection of Guidelines and Prototypes

Master's thesis in Interaction Design and Technologies

JOHAN LJUNGBERG AND ALEXANDER SANDBERG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2019

Master's thesis 2019

Designing Synchronised Multi-user Augmented Reality Experiences

A Collection of Guidelines and Prototypes

JOHAN LJUNGBERG AND ALEXANDER SANDBERG

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2019

Designing Synchronised Multi-user Augmented Reality Experiences
A Collection of Guidelines and Prototypes
JOHAN LJUNGBERG AND ALEXANDER SANDBERG

Supervisor: Staffan Björk, Department of Computer Science and Engineering
Advisor: Magnus Willner, Inceptive AB
Examiner: Olof Torgersson, Department of Computer Science and Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A user interacting with a bone in LumbAR (see section 6.1.4 for full description).

Typeset in LaTeX
Gothenburg, Sweden 2019

Designing Synchronised Multi-user Augmented Reality Experiences
A Collection of Guidelines and Prototypes
JOHAN LJUNGBERG AND ALEXANDER SANDBERG
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

With the recent advances in AR technology, there is now the possibility to create multi-user experiences that are completely synchronised between users. This thesis defines and explores the domain of Synchronised Multi-user Augmented Reality Experience (SMARE). Existing literature and applications within the area are examined and three different prototypes are created and evaluated in order to define a set of guidelines for aspects to consider when designing SMAREs. Through a process of iteratively developing prototypes and guidelines, a set of guidelines consisting of five categories with a total of 18 guidelines is defined. The results of the prototyping are AR-Showcase, AR-Racing and LumbAR which are three rather different mobile AR applications that attempt to explore different parts of the design space. The final set of guidelines is not exhaustive and rather than acting as strict rules, the guidelines should be seen as areas worth considering when designing SMAREs, and also interesting areas for further research on the subject. Even though the guidelines were developed from mobile AR applications, the goal was to make them as general as possible so that they could be used for other platforms and technologies as well.

Keywords: Interaction design, augmented reality, AR, synchronised, multi-user, guidelines

Acknowledgements

We would like our supervisor Staffan Björk for the advice and continous support. Big thanks to Magnus Willner and the people at Inceptive and Uptive for letting us use their equipment, lending their help when needed and providing a good work environment.

Johan Ljungberg & Alexander Sandberg
Gothenburg, May 2019

Contents

List of Figures

List of Tables

1

Introduction

Augmented reality, AR, combines a real environment with virtual objects that appear to coexist in the same space in real-time [7]. The accessibility of AR is at an all time high with many newer mobile devices supporting AR and the accessibility is constantly increasing, with platforms for building AR experiences, like ARKit [8] and ARCore [9], continually expanding their list of supported devices [10], [11]. While AR on mobile devices have existed for a few years now, applications are usually a single player experience or a limited multiplayer experience that is not synchronised in real-time between several users' devices. During 2018, Apple released ARKit 2 which added multi-user support for ARKit and Google released the Cloud Anchors technology, both of which lets users place virtual objects on surfaces in the real world, which can then be seen and interacted with by multiple users, through their mobile devices, in real-time [8], [12].

Creating synchronised multi-user AR applications presents challenges, issues and possibilities that are new to the domain of AR development. There are a number of interesting areas that companies and developers might need to take into consideration when designing applications with this technology in mind. Some of these areas could be what possibilities there are for interaction between users, how information can be visualised as well as different possibilities and limitations the location-based aspect of the technology presents.

As a result, the master thesis will answer the following research question:

What aspects should be considered when designing synchronised multi-user augmented reality experiences?

The planned results of the thesis are guidelines, pointing to interesting aspects that would be worth considering when developing synchronised multi-user AR experiences. A number of prototypes, exploring different possibilities within the design space, will be developed and tested as a means to identify and to some extent evaluate these different aspects.

1.1 Stakeholders

For this thesis, a number of stakeholders that affects or can be affected by the results of the thesis have been identified.

1.1.1 Chalmers University of Technology

The thesis is written at the Interaction Design and Technologies master programme at Chalmers University of Technology. The university has rules and guidelines on how to properly do a master thesis, which will affect the thesis. There will also be meetings with the assigned supervisor regularly throughout the span of the project.

1.1.2 Inceptive and Mixtive

Inceptive [13] is a consulting firm with offices in both Gothenburg and Stockholm. In addition to having on-site consultants, Inceptive has a sister company Mixtive [14] that is working with several in-house projects within the area of mixed reality. Magnus Willner, the CEO of Inceptive Gothenburg, presented multiplayer AR as an interesting area to explore, which sparked the idea for the thesis. Most of the thesis work will be carried out at their office where access to required hardware is provided and prototypes can be tested in their systems if needed. In addition to this, there are usually employees with experience within AR present that can lend their expertise if technical issues arise.

Mixtive is currently working on several AR projects and are therefore interested in the results of the thesis. AR Call [15], Spotbuild [16] and AR School [17] are some of their current projects. AR Call allows users to have a conversation with someone else in AR, where the person you communicate with is represented by an avatar. Spotbuild allows you to place blocks and build structures in a similar manner to Minecraft [18] and place them at the location you are within the real world. Other users can then go to a spot and see what has been built there. AR School aims to digitalise and enhance parts of the school for both students and teachers by providing educational AR experiences on schoolyards all over the world.

1.1.3 AR Developers and Users

The planned result of this thesis is a number of guidelines that are targeted towards AR developers, who hopefully can create better synchronised multi-user applications by taking these guidelines into consideration. In addition, the users of such applications will also be affected by the results of this thesis as the AR experience will be designed with the user in mind, hopefully improving the overall user experience of the product.

1.2 Delimitations

As described in section 2.1.1, there exists several different types of systems which can be used to experience AR. Any digital prototypes developed during this project are developed for handheld devices, more specifically mobile AR. While using a head mounted display such as AR Glasses would be an interesting approach, no such hardware was readily available at the start of the project.

The experience of co-located multi-user AR applications can differ depending on if the users are using headphones or not. If headphones are not used, sounds from a device can be heard by several users, while using headphones implies that a sound from a device only can be heard by the user that is wearing them. For the purpose of this thesis though, it is assumed that no user is using headphones.

Synchronisation is one part of the experience that this thesis will focus on which aims at updating the states of objects in real-time for all devices. This thesis will however not focus on minor latency issues and still call an application synchronised even if an objects state is not shown at all devices the exact moment it is changed.

When it comes to creating mobile AR experiences there are a few options such as artoolkitX [19], Vuforia [20], ARKit [8] and ARCore [9]. During this project, the selected platform is ARCore because of its new Cloud Anchors [12] technology which sparked the idea for this project as it allows for the creation of synchronised multi-user AR experiences. Also, at Mixtive the employees have expertise in ARCore which played a part in the decision, since this meant it was possible to get help from them if any problems occured during implementation.

2

Background

This chapter covers the brief history of AR and the different types of systems used with AR. To create AR experiences, certain tools such as Unity and ARCore can be used and these are further explained in this chapter. In addition to this, an introduction to the area of multi-user AR is made by presenting some already existing applications to help gain a better understanding of the domain.

2.1 Augmented Reality

Although the term augmented reality was coined first in 1990, some appearances resembling AR can be dated back to as early as the 1950s [21], [22]. The first real, functioning AR system, Virtual Fixtures, was developed by L.B. Rosenberg in 1992. The idea behind Virtual Fixtures was to improve human productivity by overlaying information onto the real world.

Figure 2.1: Louis Rosenberg using Virtual Fixtures. Source: [1]. CC-BY-SA-3.0.

2.1.1 Systems

AR can be experienced through different types of devices and according to Carmigniani *et al.* [22], there are three major categories of displays used: handheld displays, head mounted displays (HMD) and spatial displays.

2.1.1.1 Handheld

Handheld displays are usually integrated into a small device that produces AR by projecting virtual objects onto a video feed from a camera attached to the device [22]. The best examples of such devices are modern smartphones and tablets, which typically have the required sensors such as camera, GPS and accelerometer to compute virtual objects that are aligned to the real environment. Smartphones are extremely widespread and all new smartphone devices that uses either iOS or Android are AR-compatible [10], [11], which currently makes it the most accessible way to experience AR from a consumer perspective.

2.1.1.2 Head mounted

HMD is a display that is worn on the head, historically as part of a helmet [22] or more recently as glasses. Many believe AR Glasses to be the next step in AR development and companies are competing with different products and prototypes like Microsoft HoloLens [23], Magic Leap One [24], Glass [25] and Vuzix Blade [26], where some of them focus more on industrial use to simplify and help with work tasks while others hope to be adopted by the mainstream consumer.

2.1.1.3 Spatial

Spatial Augmented Reality (SAR) refer to technologies like holograms, projectors that are used to display virtual objects directly into the physical space [22], [27]. As a result, users can usually experience SAR without having to personally wear or carry any AR equipment which makes it a suitable system for larger exhibitions like a museum or convention.

2.2 Unity

Unity [28] is a game engine developed by Unity Technologies [29], primarily for creating 2D and 3D games, but it also provides tools for creating AR and VR experiences as well as tools used by artists and designers within other fields [28]. With support for over 25 platforms, Unity provides the ability for creators to create cross-platform applications that reach a wider audience across several platforms. The base version of Unity is free to use as long as your revenue or funding does not exceed a certain amount. Beyond the base version there are subscription based versions that can be used for projects with higher revenue that also provide some additional features and support.

Unity supports C# for writing code in the form of scripts and for Windows and macOS the default integrated development environment, IDE, is Visual Studio IDE [30]. Unity also provides free online resources such as tutorials, courses, forums and documentation that can ease development for beginners aswell as more advanced users [31]. The large user base for Unity means that there are many people that have experienced the same problem and therefore it is easy to share knowledge and learn from other users through the forums.

2.2.1 Networking

Unity provides their own solution for creating multiplayer games called Unity Multiplayer [32], commonly referred to as UNet. UNet can be used to create real-time multiplayer games and provides services such as matchmaking and servers localised in several parts of the world. UNet can be used for free for up to 20 concurrent users, with several options for potentiallty scaling up, getting more expensive the more data that needs to be transferred per second. UNet operates on a client-server basis, where one of the clients acts as a host. UNet is being deprecated in the next big update of Unity, where it is being replaced by their new networking solution.

Another option for creating multiplayer games is PUN [33], a third party solution from Photon [34]. PUN provides much of the same functionality as UNET, such as matchmaking, servers in several parts of the world and a free plan for up to 20 concurrent users. It is well documented and provides easy to use tutorials and examples for beginners. Similar to UNet, it operates on a client-server basis, but uses dedicated cloud servers for hosting.

2.3 ARCore

ARCore is a platform by Google for developing AR applications for mobile devices. ARCore uses markerless AR by using motion tracking that understands and tracks the phone's relative position to the world, and by detecting size and location of horizontal, vertical and angled surfaces [9]. In addition to this it also estimates the lighting conditions in the real environment which can be used to apply more realistic lighting to virtual objects placed within the real environment. The fact that ARCore uses this type of location-based, markerless AR allows the user to place an object on a tracked surface and then move around it to view it from different angles and even allows the users to turn around and walk away, later returning to the object in the same place that it was placed [9].

ARCore is supported by a wide range of Android devices running Android 7.0 or later [10]. It is also supported by a number of iOS devices that are compatible with ARKit [11] and running iOS 11.0 or later. Google provides an ARCore software development kit, SDK, which can be used for developing ARCore apps in Unity.

2.3.1 Cloud Anchors

Cloud Anchors is a feature provided by ARCore that can be used to create synchronised multi-user experiences, even across Android and iOS [12]. In ARCore, an anchor refers to an approximation of a fixed location and orientation in the real world, which are automatically adjusted as ARCore improves its understanding of the environment while the user moves around [35]. Cloud Anchors are anchors that are hosted in the cloud, so that multiple users that are in the same physical area can use the same anchor to establish a common frame of reference in the environment across several devices.

2.4 Multi-user AR Applications

Many applications can be considered to be within the domain of multi-user AR. In this section, a few quite different applications are presented which to a varying extent may or may not fit within this area, but are important either way to gain a better understanding of the domain.

2.4.1 Just a Line

Just a Line [2] is a mobile AR application in which the user has the ability to create virtual drawings within the real world. The interactions with the app is simple, to

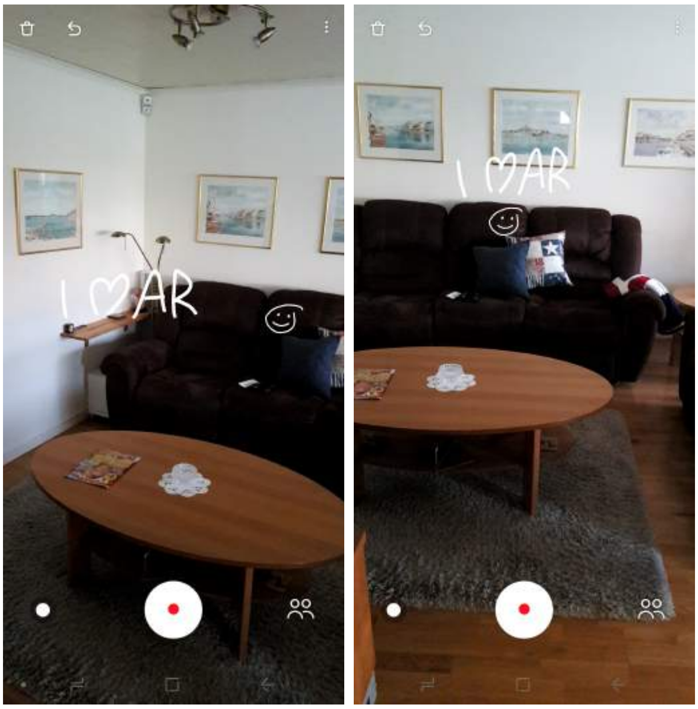

Figure 2.2: Same drawing viewed from two different angles in Just a Line [2].

draw a line just touch the screen and either move the finger on the screen, move the phone or both. The line is then saved in the location it was made so the user can walk around and look at it from different angles. If the user is proud of the work or just want to show it to some friends, they can use the application's built-in recording functionality to create a video of it.

The application allows one user to create drawings alone, or two users to collaborate. If two persons decides to draw together, Just a Line uses the Cloud Anchor technology [12] to create a cross-platform, synchronised multi-user experience [2]. The users pair their devices by just standing side by side looking at the same spot.

2.4.2 ARrrrrgh

ARrrrrgh [3] is an AR game available on both iOS and Android. It is a turn-based game for two players which is played on one device. When the game starts, one player takes the device and chooses a good hiding place in the real world. The application then digs a hole at this spot in AR and hides a virtual chest inside that hole. After that, the device is handed over to the second player who is going to find the hidden chest. To their help there is a bar at the top of the screen that hints about how close the player is to the chest's location. When they think they have found the spot, the player presses a button to dig a hole at the location and if it was the correct spot, the chest will be dug up, otherwise they can continue the search. This is repeated until the chest is found.

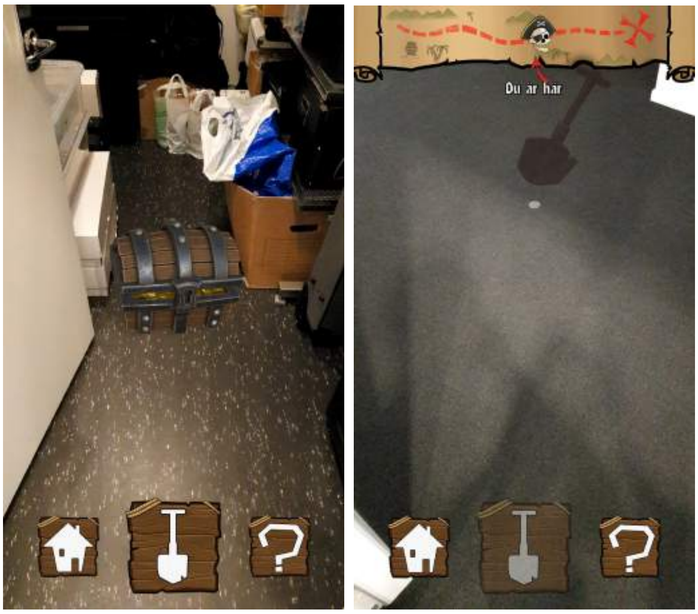

Figure 2.3: Two screenshots from ARrrrrgh [3]. The left screenshot shows the hiding phase, the right shows the seeking phase.

ARrrrrgh takes the old hide-and-seek and hot-or-cold game and brings it into an AR application. It has simple interactions which only requires the players to press a button to choose a location, and makes good use of the tracking ability that the phones of today have. ARrrrrgh manages to create a multiplayer AR experience by letting users take turns on the same device.

2.4.3 Pokémon GO

Pokémon GO [4] is a location-based game for mobile phones. The purpose of the game is to find and collect all different kinds of Pokémon that are hiding in real-world locations. To catch them, the player has to be at the location of the Pokémon and then throw Pokéballs at it. Every thrown Pokéball has a certain chance to catch the Pokémon, and it is dependant on several parameters like for instance the type of ball, how the player throws the ball and where the ball hits the Pokémon. If the Pokémon breaks free from a Pokéball, there is a chance that it will run away and the player has failed to catch it. Catching Pokémon can be done in two different ways, as illustrated in figure 2.4; either the Pokémon will show up in the middle of the screen without AR, or an AR-mode can be activated which will require the player to look around in their environment to find the Pokémon. There is no differences between the modes in terms of gameplay, it is just an aesthetic feature.

Figure 2.4: Two screenshots from Pokémon GO [4]. The left screenshot shows catching a Pokémon in AR, the right shows the same thing without AR.

The Pokémon caught can be used for different things. One thing is that they can be traded to the virtual Pokémon professor in exchange for candy, which is used for evolving Pokémon from one type to a more powerful one. Players can also trade their caught Pokémon with each other. Pokémon can also be used for different kinds of fighting. Either the players can battle each other for training, or battle at a gym. There are three different teams in the game that the players join when they create an account; a red, a blue and a yellow team. These are also the teams that are fighting in the gyms to control them. Only one team can control each gym, and that team can place a few Pokémon in the gym to defend it. The other two teams can go to the gym and try to take control of it by fighting the Pokémon in it. There is also another type of fight that happens at the gyms called Raid Battles. In Raid Battles, Pokémon will spawn in the gyms at random times, and they are so powerful that it requires many players in order to beat it. If a group of players manage to beat the Pokémon, they will have the chance to catch a less powerful version of it themselves.

Pokémon GO is a game which definitely can be classified as a multiplayer game since there are several ways of interacting with other players in the application, through either trading, fighting or sending gifts. It is also possible to group up and take down a Pokémon together in a Raid Battle. But there are a lot of things in the game that does not require other players. Catching Pokémon can be done alone and collecting items such as Pokéballs can be done at Pokéstops which are placed on different locations on the map. The one and only feature that uses AR can only be used in a singleplayer context, which means that even though Pokémon GO is a multiplayer game, and also an AR game, it is probably not an multiplayer AR game.

3

Theory

This chapter covers a definition of AR and where it exists within the area of mixed reality. A definition for synchronised multi-user AR experiences is also presented to make a distinction between different types of multi-user AR. Further, the concept of wicked problems is described and some guidelines for AR are presented, as an understanding of both these is important to reach any meaningful results.

3.1 Wicked Problems

Problems can be seen as wicked or tame problems. According to Rittel [36], a tame problem is a problem which can be solved if enough information is given about it. It is also easy to say if the solution is right or not. Examples of tame problems are math problems. If enough information is given, it is possible to find a formula and calculate a correct answer.

All problems are not possible to solve. As mentioned by Rittel [36], wicked problems are problems that does not have one perfect solution. Instead, people can judge a solution as good or bad, but it is not possible to say whether it is right or wrong. Rittel [36] also states that a wicked problem can be treated in many different ways, and that it is possible to work on a solution forever since there is no point where it is completely done.

One kind of problem that can be seen as wicked are design solutions [37]. A design can be either good or bad, but it is not possible to get the design that is perfect to everyone. Instead a solution to a design problem should be worked on until it is satisfying enough, which is typical for a wicked problem [36].

3.2 Augmented Reality

Augmented reality, AR, combines a real environment with virtual objects that appear to coexist in the same space in real-time [7]. AR can refer to augmenting any sense such as sight, hearing, touch and smell but most commonly refers to augmenting the sense of sight, which is also what will be referred to by this report when mentioning AR, as indicated by the definition above.

In the general area of mixed reality, AR is just one part of the continuum of real-to-virtual environments [7], [5]. One end of the continuum is entirely real environments and the other one is entirely virtual environments (Virtual Reality). In between is AR and Augmented Virtuality, in which real objects are added to virtual environments.

Figure 3.1: Reality-virtuality continuum. Adapted from: [5].

3.3 Synchronised Multi-user AR Experiences

Previous to Cloud Anchors and similar technology, multi-user AR applications and multiplayer AR games did already exist. Pokémon GO is commonly referred to as an AR game, even though using AR is optional and only applied to a small part of the game. In addition to this, it could also be considered a multiplayer game, although mostly asynchronous. As mentioned in section 2.4.3, although Pokémon GO is a multiplayer game, and could also be considered an AR game, the AR feature is single-player only. Catching Pokémon can be considered multiplayer in the sense that several players can see the same Pokémon appear on similar locations of the map but the action of one user does not update for other users in real-time while in AR and trying to catch the Pokémon, states are thus not synchronised. Software such as Google Drive [38] and Trello [39] is synchronised in the sense that the actions of one user is updated in real-time to everyone else they are currently collaborating with. Other examples are some of the many real-time multiplayer games such as FIFA [40], Fortnite [41] and World of Warcraft [42].

So while applications such as Pokémon GO is sometimes referred to as a multiplayer AR game in the broadest sense of the definition, this is not the domain which this thesis will focus on. As such, it is necessary to make a distinction by defining what domain this report will focus on. Using Pokémon GO as a counter example, this thesis will focus on *Synchronised Multi-user AR Experience* (SMARE). SMARE refer to applications where the state of virtual objects are synchronised within the real environment between several users' devices in real-time, such as Just a Line. The term multi-user is used in this definition over multiplayer as the term multiplayer is usually used when talking about games, while multi-user is a broader term that covers other areas than games, while not necessarily excluding games.

SMARE implies that users will usually be co-located, as the virtual objects that are synchronised between users are placed in the physical environment and therefore users need to be nearby each other to see the same object. There are of course exceptions to this and one example would be if two people on completely different locations in the world looked at the same virtual object placed on the moon or something similar. Bergström [43] writes about togetherness and how it can be facilitated in game design. Even though game design is not the focus of this thesis, the report contains interesting parts about the concept togetherness which can be useful for areas outside of game design aswell. Bergström defines togetherness as *the positive emotions experienced by people doing things together in groups* which is a definition that also can be used when talking about the concept of SMARE and co-located users.

3.4 Augmented Reality Design Guidelines

Google presents their Augmented Reality Design Guidelines [44] which aims to provide best practices and guidelines for designing mobile AR experiences. These guidelines are designed with ARCore in mind but provide mostly general guidelines which can also be applied to mobile AR experiences that are using other similar platforms. The provided design guidelines cover the following areas: *Environment*, *User nuances*, *Initialization and adding virtual assets*, *Interaction with virtual assets*, *Designing the experience* and *Realism*.

The *Environment* section mentions things to take into consideration concerning the real-world and augmented environment, such as where the application will be used, what are good lighting conditions and detectable surfaces [44].

The *User nuances* section covers considerations concerning User movement, User comfort and Reducing user frustration [44]. Some of the aspects to consider that are brought up in this section are what different types of movement the experience is designed for and how well the application accommodates these, as well as the general comfort of the user when using mobile AR for a prolonged period of time.

Initialization and adding virtual assets covers the areas of Plane discovery, Optimal placement range and Placement, mentioning how to provide meaningful feedback, visual distinction and intuitive interactions for these aspects of the AR experience [44].

In *Interacting with virtual assets*, guidelines are presented for how to handle some different interactions between the user and virtual objects [44]. The interactions covered by this section are Selection, Translation, Rotation and Scaling. In addition to these specific interactions, some general guidelines concerning gestures and proximity are also presented.

Designing the experience covers UI components and Experience, which provide some best practices when designing a user interface for immersive AR experiences as well as several user experience, UX, guidelines for an overall positive user experience from start to finish [44].

The *Realism* section mentions things such as depth, shadow, lighting, physics and how these can be used to make the AR experience seem more realistic by simulating object presence [44].

3.5 Related Research

There have been some interesting previous research within the area of multi-user AR which was studied to get a better understanding of the domain, identify interesting areas within it and see what have previously been done.

3.5.1 Mixing Realities in Shared Space

Billinghurst *et al.* [45] presents the Shared Space project, with the aim of investigating how AR enhanced with different interfaces can be used to develop effective face-to-face collaborative environments. Shared Space integrates and explores a number of interface technologies that are defined as *Augmented reality*, *Collaborative computing*, *Physical interfaces*, *Spatial 3D user interfaces* and *Computer vision tracking and registration*. Further it is noted that while regular graphical user interfaces are considered direct, in the sense that the user manipulates interface object similar to how one would with physical objects, they are simply metaphors used to improve a users understanding of how to interact with the interface. Shared Space investigates the use of an interface that combines spatial, physical and tangible interactions in an AR setting designed for collaborative work.

The key aspects of Shared Space, as described by Billinghurst *et al.* [45], are *Augmentation*, *Collaboration*, *Interaction* as well as *Tracking and Registration*. Users wore a HMD that would show them the real environment through a camera and also recognise physical cards with specific patterns, on which virtual objects would appear. Since all co-located users shared the same interface, they could simultaneously see the same virtual and physical objects which allowed them to work together while the fact that users could see each other's facial expressions and body language supported natural face-to-face communication. Spatial relationship between markers and physical interaction like rotating and shaking could be used to trigger interactions with the virtual objects, such as placing a virtual alien and an UFO next to each other would result in the alien flying inside the UFO.

A number of applications were developed, explored and tested to varying extents. The conclusion was that people had no difficulties in using the AR interface and behaved similarly to regular face-to-face collaboration with physical objects [45]. The physical and tangible aspect of the AR interface made interactions and collaboration very easy and intuitive.

3.5.2 Multiplayer Collaborative Training System

Yangguang *et al.* [46] conducted an experiment in which they developed a multiplayer collaborative training system, MCTS, in AR. They built the system as an escape room, in which the participants need to solve different tasks in a closed room in order to get out. All clues of how to solve these tasks was located in AR and the only tool that the participants had was an iPad each. Yangguang *et al.* [46] then did user tests to find out how good these kind of systems are compared to the traditional physical training systems. They made a physical and a virtual environment with the same tasks and the same difficulty. The users were split up into an AR and a non-AR group. Those groups were divided further into a group were everyone knew each other, a group were everyone had met a few times before and a group were no one knew each other. When the groups tried the escape rooms, Yangguang *et al.* [46] found out that the average completion time was lower for the groups that solved the tasks with help of AR. Therefore they came to the conclusion that MCTS could improve the problem-solving ability among the players.

4

Methodology

This chapter covers methods and processes that are useful for doing design work and research in an efficient manner. In addition, agile principles are presented for guiding implementation of digital prototypes.

4.1 Research Through Design

According to Frankel & Racine [47], there are three different design research strategies; research for design, research through design and research about design. Research through design is about learning from doing actual design work, and can be based on result from previous works or nothing at all. This area is popular and growing rapidly since both researchers and practitioners are coming up with new information about the subjects [47]. When doing research through design, the process generally includes three different phases; research phase, design phase and publication phase [48]. In the research phase, information about the interesting area and previous works are studied. In the design phase the actual design work is done, and it can be based on the results that was found in the research phase. At last in the publication phase, the results are presented. The results should be both a published text and some kind of artefact that has been designed. As proposed by Gaver [37], the design results can be grouped together into *annotated portfolios*, which can help designers that are trying to solve similar problems in the future.

4.2 Literature Review

Literature review is a method essential to properly researching a topic. Reviewing research literature helps to establish an understanding of an area, what has been done before within that area and whether a topic is new as well as helps formulating a research question and framing the problem [49], [50], [51, pp. 112].

4.3 Iterative Design

The design process does not always look the same. Moggridge [52] proposes a structure that can be used for design processes, which is built up by ten different steps; constraints, synthesis, framing, ideation, envisioning, uncertainty, selection, visualisation, prototyping and evaluation. These steps are then iteratively repeated throughout the design process, even though the order can vary from time to time.

Dam & Siang [53] writes about "5 Stages in the Design Thinking Process", where the process starts with an empathise phase, which then is followed by define, ideate, prototype and test phases. For interaction design, Preece *et al.* [54] says that the process includes four basic steps; establishing requirements, designing alternatives, prototyping and evaluating.

All the processes described above are iterative processes, which are common when designing. According to Moggridge [52], iterative processes are the best and fastest way to get a good design. Most processes includes the steps ideation, prototyping, testing and evaluation in some way, but it is not uncommon that they also have steps like research or forming requirements in the beginning. Research is important to do early in the process to minimise the number of changes that is needed later in the project [55].

4.3.1 Ideation

To come up with ideas for a design is not always easy. A common method to use to make it easier is brainstorming, in which one or more persons tries to come up with ideas and evolve them [56]. Even though brainstorming is popular and have been used for a long time, Diehl & Stroebe [57] names some obstacles that can lower the productivity, especially if done in a group. The obstacles named are *production blocking*, *evaluation apprehension* and *free riding*. Production blocking is when a person has an idea, but can not say it because others are talking and then the idea is forgotten. Evaluation apprehension is the fear one person have of getting negative feedback on an idea. Free riding is when a group member does not contribute that much and instead rely on the effort of the rest of the group, because the result is not individual anyway.

Today, there exists many methods that help designers through the ideation phase. They can be very different, but many of them have a similar purpose; to give some kind of input to stimulate the brain of the designers to help them come up with ideas that they otherwise would not have thought of. It can be valuable to designers that are stuck in their thinking to give them a push in another direction, but also to come up with new unique ideas. Examples of methods that can be good to use for these purposes are sketchstorming, bodystorming, creative toolkits and cheatstorming [51], [58].

4.3.2 Prototyping

As stated by Moggridge [52], one should prototype early and often with each iteration taking the prototype a step further. As described by Martin & Hanington [51, pp. 138], prototypes can be divided into different subcategories referring to their level of fidelity.

Low-fidelity prototypes are useful in the early stages of a project such as the ideation process, to evaluate concepts and ideas to see if they are worth developing further or just to build a better understanding of a concept among team members [51], [59]. Low-fidelity prototypes are primarily used for an internal development purpose but can also be used for early testing of ideas on users to identify core issues and get feedback in the early stages where making changes and iterations is usually faster and cheaper. Low-fidelity prototypes can many times be physical and Lankoski & Björk [49] presents how paper prototypes can be used as a low-fidelity prototype when prototyping games but also emphasises that not all functionality can be tested in a realistic way. Paper prototyping can be useful to simulate certain aspects of a design and give feedback on those while other aspects are harder to simulate using a pen-and-paper approach.

High-fidelity prototypes are more polished and closer to the final product when it comes to appearance, look and feel, and even basic functionality [51]. This type of prototype can be used in the later stages of project for testing on external users to receive feedback that covers additional aspects of the product such as aesthetics, interaction and usability. High-fidelity prototypes are often digital and may allow for testing of functionality that might be limited when testing on physical prototypes [60].

4.3.3 Evaluation

In an iterative design process it is important to evaluate the prototypes that are being made, to get feedback and to identify the strengths and weaknesses of the design. Evaluations can be done in different ways and the test methods that are used will determine what type of data that is gathered. The data collected can be categorised as either qualitative or quantitative [49].

Quantitative data can usually be summarised in numbers, like "9 out of 10 liked the product" or "85% of the test subjects completed this step", and as such it is possible to gather great amounts of quantitative data and still summarise and analyse the results with little effort and in an unbiased manner [61]. Quantitative data can be used to effectively point towards parts of a design that are good or bad, while it can be more difficult to fully understand what the underlying issues are for a problem or why one certain feature is used more. Examples of test methods that will give quantitative data are questionnaires with closed-ended questions, A/B Testing and evaluative research [51].

Qualitative data is more rich feedback than quantitative data and consists of for instance descriptions, observations and opinions collected from the tests. As stated by Lankoski & Björk [49], qualitative data is gathered in order to focus on meaning and understanding the things that are tested. Gathering, analysing and summarising qualitative data generally takes more time than quantitative and also focuses on a smaller sample size. The conclusions that are drawn from analysing qualitative data can to some extent be biased as they will depend on a subjective interpretation of the data [61]. Some methods that can result in this type of data are interviews, heuristic evaluation, think-aloud protocol, observations and affinity diagramming [51].

4.4 Agile Development

Agile is a collection name for several different ideas of what the structure of a software development project could be, and these ideas are written in the Agile Manifesto [62]. Then there are different methods that applies to these ideas like for instance Scrum [63], Extreme Programming [64] and Lean [65]. The different methods connects to the agile ideas in different ways through their own values, principles, roles, practices and artefacts [66]. These methods can be mixed freely in different ways to get a development process that suits the team as good as possible, so there is no need to choose one method and ignore the rest.

4.4.1 Scrum

Scrum is one of the agile methods, which contains several famous practices. The Daily Scrum meeting is one, which is a meeting that should be held once a day where all team members quickly tells the rest of the group what they did yesterday, what they are going to do today and if they ran into any problems [67]. The Scrum Board is another, which is a board that keeps track of the different tasks of a sprint and who is working on what.

Working in sprints is typical for a team that uses Scrum. A sprint starts with some planning, where the team decides how much work that should be done until the end of the sprint. Then the team works on these tasks until the end of the sprint. The length of a sprint can vary but it is usually somewhere between one week and one month [68]. When a sprint finishes the team does a retrospective review where they discuss how things went.

5

Process

The iterative process of this thesis included several steps and started with a prestudy. After that, three iterations consisting of ideation, prototyping and evaluation were done, where each iteration focused on a unique concept for the project. During every iteration a prototype was developed, together with new guidelines and refinements of previous guidelines. In this chapter, the working process is described and design choices that were made during the iterations are explained.

5.1 Prestudy

Before starting with the prototypes, the first weeks of the project were dedicated to learning and getting a better understanding about the subject of SMARE and related areas. This was mostly done through literature review by reading about the existing research from different databases on the internet. Databases that were used the most to find information was Google Scholar [69], IEEE Xplore [70], DiGRA Digital Library [71], ACM Digital Library [72] and Chalmers Library [73]. The following keywords were the ones that were used the most when searching for relevant information in the databases.

- Augmented Reality
- Mixed Reality
- multiplayer
- multi user
- co-located
- shared space
- collaboration
- real time
- synchronous
- asynchronous

The keywords were used both alone and in combinations with each other in order to find relevant articles and papers. When the literature found covered interesting areas, the references of that literature was checked in order to find more interesting literature.

Beyond searching online databases for information, literature from previous courses were used as well. Since that literature had been used before, the content of them was known which made it easy to find information relevant for this project. A number of AR applications were downloaded and tested in order to get a good understanding of what is existing on the market today. All applications were not SMARE applications, but they all were helpful when narrowing down the scope of SMARE.

5.1.1 Planning

The planned results of the thesis were guidelines for people working with SMARE. First of all, literature review and background research within the area was done as described in 5.1. From here, the plan was to iteratively design three prototypes, investigating different areas of the the design space, resulting in three iterations of the design guidelines. Different methods, frameworks and processes, as presented in chapters 3 and 4, were planned to be used to reach those guidelines in a structured and correct way.

In addition to the continuous writing of the report, the plan for the final part of the thesis work was to evaluate the results of the research and the implementation, to establish the final guidelines and finish the report.

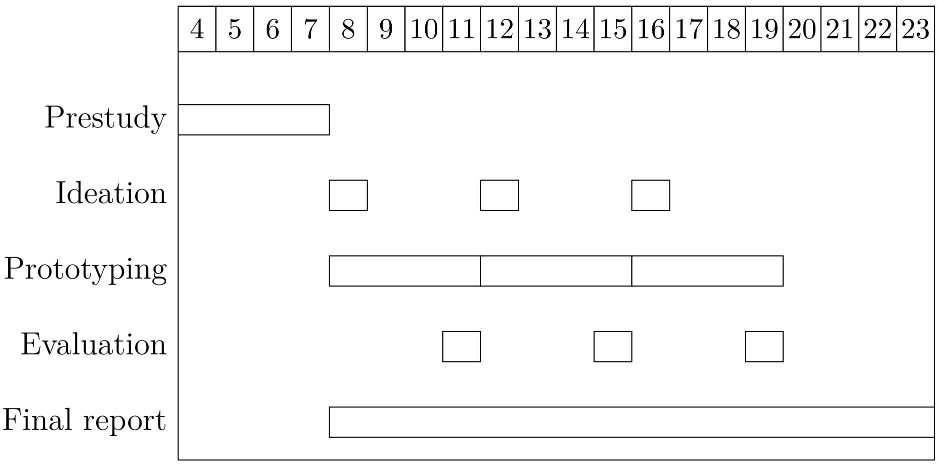

Table 5.1: Gantt chart representing the preliminary time plan of the thesis work on a weekly basis.

5.1.2 Initial Guidelines

The guidelines below were written before the first iteration of prototypes and are therefore mostly based on other guidelines that was found during the prestudy. They should be considered more like categories for guidelines than actual guidelines at this stage. These categories are a starting point that is far from finished, but they will be updated after each iteration when more practical experience is gained from the prototypes. At this time, the categories are derived from other related areas like single user AR rather than SMARE. The plan is to explore these categories in SMARE through our prototypes, and as practical experience is gained find guidelines specifically for SMARE.

Social Interaction: Mobile AR presents some unique aspects when designing for social interaction. Consider how interaction between users can be used to augment and support real world tasks while also having the possibility of keeping the presence and face-to-face interaction with other people. This guideline is based on the framework provided by Chang *et al* [74].

User Safety: An AR-application should be safe to the user. Things like walking backwards or doing large and sudden movements can be dangerous to the user if there are obstacles or other persons close. So be aware of the areas where the application is intended to be used and which interactions that it encourages the user to do. This guideline is based on other guidelines from Google and Apple [44], [75].

Play Area: Think of how big area the user will need to use your application, and how that affects your design. Consider if the application should be responsive to the actual play area and for instance scale objects to fit the physical world. This guideline is based on information from Google's AR design guidelines [44].

Accessibility: Considering the possible constraints of the user and environment, there might be situations where the user is not able to move around. Providing alternative ways to to partake in the experience, such as rotating and moving objects closer, increases accessibility. Visualise information such as text and instructions in a way that they are visible from any angle. This guideline stems from Google's AR design guidelines [44].

User Comfort: Consider that users can get fatigued after using AR for a while. Try to avoid forcing the user into long sessions by finding stopping points to allow for the user to take breaks. Allow users to pause or save progress and make it easy for them to continue, even if their physical location is different. It can be tiresome to hold a phone for extended periods of time, encourage users to move their phone around in order to change the position in which they are holding it. This guideline is based on the guidelines provided by Google [44].

Feedback: Think of the feedback that the application gives the user after different actions. Providing for instance audio feedback on the user's own or other users' interactions can enrich the AR experience. This guideline is based on what Apple and Google writes in their own AR guidelines [44] [75].

5.2 First Iteration

This section goes through the ideation, prototyping and evaluation phases of the first concept, and how this led to the first iteration of guidelines.

5.2.1 Ideation

The first iteration started with an ideation session. During the planning stage, it had already been decided that the first prototype would be an application which would focus on exploring the dynamics of having one master user who administers the experience for several other participants who would engage by watching the experience through their own devices.

The ideation started out with a brainstorming session, with the purpose of coming up with ideas of a concept for a "master user and observer" application. The basic concept that was developed during the session was an application where one user would be able to upload 3D models and show them in AR for other people. The thought was that this application could for instance be used in a school class, where the teacher could show models of something that they are currently working with, and the students could have their own devices and take a closer look of the objects. Another scenario would be during a meeting at work, where one person is holding a presentation and uses the application to showcase a model of some sort right at the conference table.

After the basic concept had taken shape, other things like the user interface was discussed. How should the user add models to the AR scene, and in which way should the user interact with the objects already placed? Two different interfaces for solving the problem of adding models to the scene came up during ideation, one in which the user dragged objects from a 2D list into the 3D space and one where the user tapped in the 3D space and different models that could be placed would appear besides it in the real world. After some discussions it was decided that the prototype should use the 2D menu, partly because it felt more natural than the other option and partly because Google's guidelines for AR [44] recommended this type of interaction.

When the brainstorming session was done and an initial concept had been created, bodystorming was conducted. The reason for this was to identify any issues and potential details that was not considered during brainstorming. Therefore, two scenarios of using the application was role-played using mobile phones as props. No missing parts of the concept were identified and no additional ideas were conceived as a direct result from this.

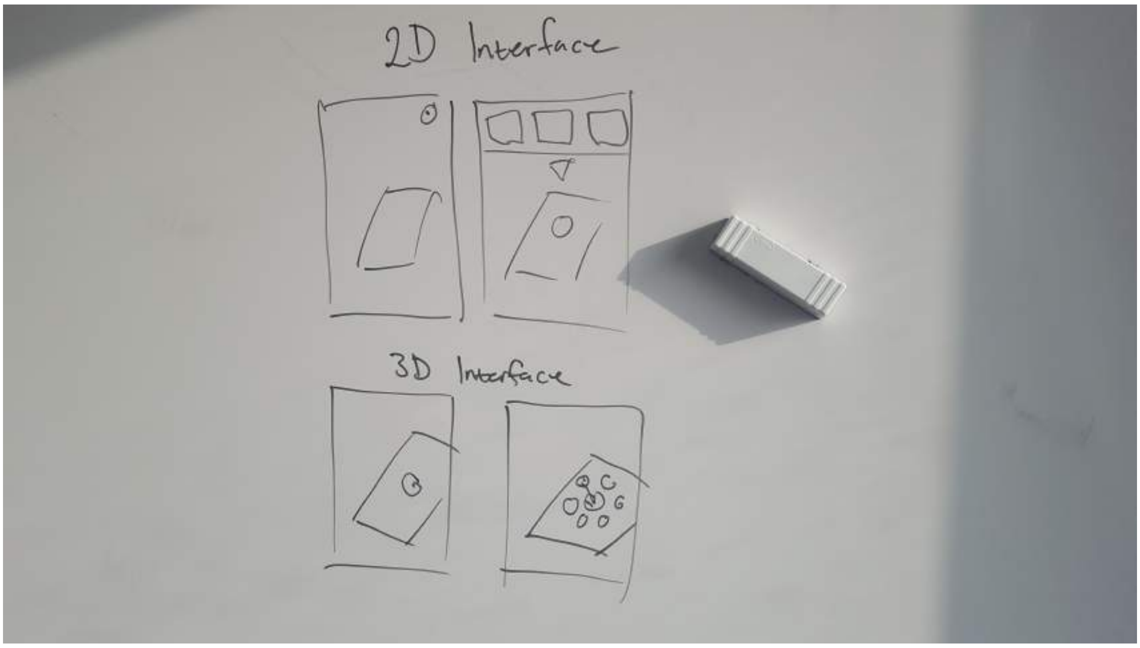

Figure 5.1: Some initial sketches of selecting a model using either a 2D or 3D interface.

5.2.2 Prototyping

As soon as the initial concept was established, the goal was to create a quick prototype for some internal testing and evaluation of the concept. As such it was decided to create a physical prototype, more specifically a paper prototype. While testing things directly related to 3D and AR would be difficult with a paper prototype, it was possible to test most of the interface and 2D interactions between the mobile screen and the user. A mobile phone was used as the frame and all the UI elements were made as paper cutouts which could be moved around and replaced on the screen as the user interacted with the interface.

While testing the paper prototype, some flaws in the design were brought to life and the prototype gave some ground for discussing design decisions to solve these issues. For example, once the user had opened the top panel to place a model in the real environment, there was no way to close the panel again. This led to a discussion of what happens after a model has been placed; should it automatically close, always stay visible or give the user a button to close it down? Ultimately it was decided to give the user the option to close the panel manually, as keeping it up all the time would go against the AR guidelines of non-intrusive UI and automatically closing the panel every time a model is placed would interrupt the user's flow in the case that they wish place several models. As the final step of the physical prototyping, a quick demonstration of the concept, using the paper prototype, was also made to the advisor at the company, who did not find any immediate flaws with it.

Once the physical prototype was done and tested, it was decided to start working on the digital prototype. Implementation of the prototype followed agile principles and

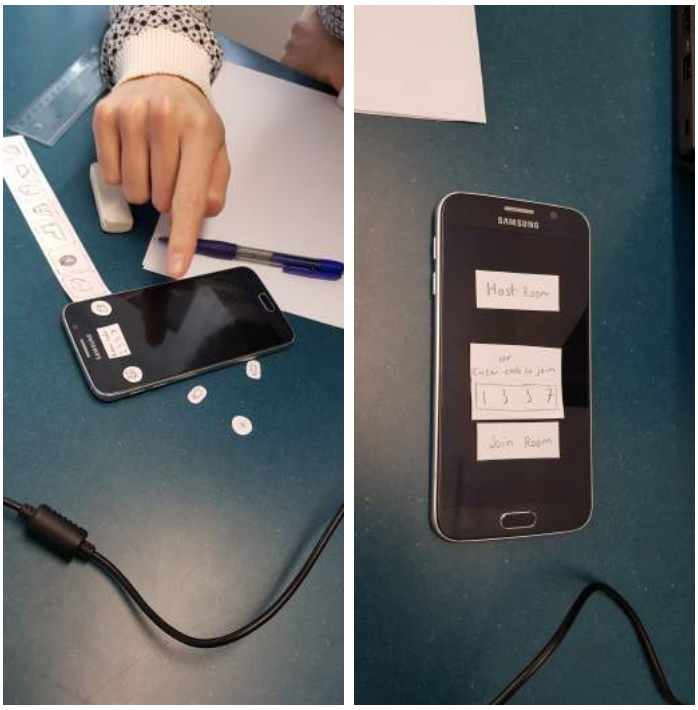

Figure 5.2: Testing the paper prototype using a mobile phone and paper cutouts.

parts of Scrum, so the first step was to create a backlog of tasks and features that should be in the first prototype. These were then prioritised and each week some of the tasks was assigned to a new sprint. The backlog was iterated on continuously as development went on and requirements changed as a result of issues, insights and design decisions that occurred along the way.

Implementation of the digital prototype started from the cloud anchors sample app provided by Google, as this gave some code examples which would help speed up development. While the sample app used UNet as their solution for networked multiplayer, the decision was made to instead use PUN, as one member of the group had previous experience in using it and UNet was marked as deprecated as it was about to be replaced by Unity and as such it seemed futile to spend time learning it. Networking was implemented such that one user will create a session, called a room, and a random 4-digit code will be generated and displayed to the users in the room. The code is then used by other users to enter the same room. Since the nature of the experience requires all participants to be co-located, using a short code that could easily be shared vocally seemed like a convenient solution.

Additionally, once a user is in the room, there is also an option to leave the room. Exactly what leaving the room meant for both roles was something that had not been thought through entirely in the ideation stage. The default behaviour of PUN was to transfer the master role over to another user in case the current master disconnects. This did however not really make sense for this app, since the idea is that one specific person starts the experience to specifically present something. As such, the decision was made to let observers leave and join as they wish, while closing the room in case the master user leaves. Also since this meant losing the current state of the room, a confirmation dialog was added for the master user to confirm leaving the room, while this was not deemed necessary for observers as nothing would be lost.

Figure 5.3: Initial interface with leave room button, room code and cloud anchor sample app onboarding.

Simultaneously as the basis for a multi-user experience was set up, work was made towards the ability for users to place 3D models in the real world. It was decided that there would be a horizontally scrollable panel at the top of the screen, from which the user can drag objects into the AR space. In the beginning, it was desired to represent the different objects in the list with actual 3D models. Unfortunately, this idea was scrapped after spending quite some time trying to figure out a technical solution without success. Instead the objects had to be represented by 2D pictures of the object. For the user the result looks almost the same, but the negative thing with this is that every object not only requires a 3D model, but also a 2D picture. So for every object that is added to the application, it is needed to take a picture of the object, then remove everything around it so the background became transparent. If it had been possible to show 3D models in the list, these extra steps could have been skipped and time would be saved. Also, if this prototype would be developed further and released to the public, people would probably want to upload and use their own objects instead of using just a set of predefined objects. Then the requirement of having a 2D picture of the object would put extra work on the user.

There was an idea in the beginning of how the placement and movement of 3D models in AR should work and how to visualise to the user whether the chosen location is valid or not. The idea was similar to a solution shown in Google's AR guidelines, where a model is floating in the air at the point where the user is touching

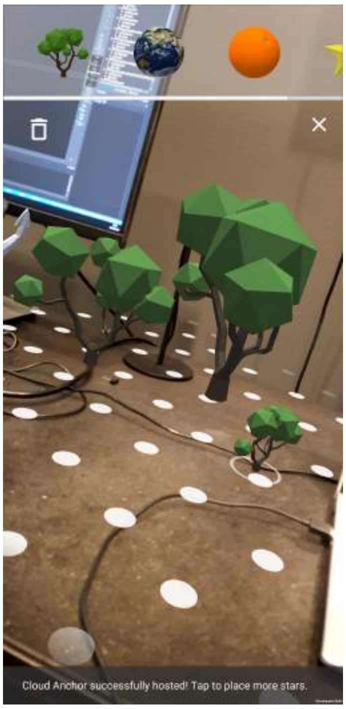

Figure 5.4: Horizontally scrollable panel containing models that can be placed in the environment by dragging them to the visualised plane.

the screen, and when the user releases their finger the model would drop onto the area beneath it. If there is no area under it, the model will be moved to the closest valid position. Holding the model above an invalid position would also turn a circle around the model red, to tell the user that it is not possible to place the model there. Even though this functionality would be nice to have in the prototype, it was given low priority. It was expected to be pretty time consuming to implement and not really worth the time as long as there was other more important stuff to do. In the end, there was not enough time to implement that solution, so instead the final solution became a variant of this, where the model stops following the finger and stays on the last valid place when the user tries to drag the model to an invalid position.

The way that cloud anchors work is that one user hosts an anchor at a detected plane in the real world, this is then hosted to a cloud server and other users can resolve a specific cloud anchor, which is identified by a unique id, by looking at the same position where the anchor was hosted. As the project started from the cloud anchor code example, the baseline was that the host taps a surface to place the anchor there. However, for the app which was being developed, the goal was to completely abstract the concept of anchors to the user as it was deemed as unnecessary information. Instead, the idea was that it would work behind the scenes without the user having to be aware of the underlying technology, by hosting an anchor at the position of the first model the user placed. After working on the implementation of this and playing around with it for a while, it was realised that it would cause some issues. Since a cloud anchor is stuck in the position in which it is placed while models placed by the user can be freely moved around, it would cause a disconnect between where the cloud anchor and the model is placed in the real world and as such very difficult for users to identify where they need to look to start the experience. So instead, it was reverted back to how it initially worked but with a change from an anchor to a more fitting metaphor. Instead a 3D model of a compass was chosen as it would be used as the origin of the experience.

Figure 5.5: Initial representation of the cloud anchor.

Following Google's AR Guidelines, specific touch gestures were added to rotate, translate and scale objects. While reading up on the subject, a free Unity plugin called Fingers Lite [76] was found which made it really simple to handle different gestures made by the user such as tap, pan, rotate and more. Moving an object is done by simply dragging a model around on the plane and releasing to place it. Scaling objects is done using two fingers to do a pinch motion. Rotation can be done in two different ways, as was suggested by the guidelines previously mentioned. A model can be rotated using either one or two fingers. When using two fingers the user rotates the fingers in a circle motion, or the user can use a swiping motion when using one finger to rotate.

While adding rotation, scaling and translation of models was pretty straight forward, a discussion was brought up on whether everyone should be able to individually rotate a model, following the *Accessibility* guideline presented in section 5.1.2, or not. In the end, it was decided that only the host would be able to rotate models, as otherwise it could diminish the experience of SMARE and the feeling of seeing the same objects in the same manner if everyone had their own rotation on them.

In addition to adding and manipulating models, the master user can also delete already placed models. The button to delete an object only appears first after the user has selected an object. Once an object is selected, the user can either click the trash bin button, or drag the object towards it. While clicking the trash bin would probably be enough, it was important to give the users different options depending on what they find most intuitive, as dragging an object to the trash bin is similar to the interactions of dragging an object to place it and also move it.

To improve the general AR experience, a few different approaches were considered. First of all, virtual objects are not generally occluded by real objects which can greatly decrease the immersion by having objects appear where they are not and also cause a lot of clutter if objects appear to be on top of each other when in reality there is a table or some other real object in between. To improve this issue, a plugin called ARCoreUtils [77] was added and modified in order to add occlusion between detected planes. Secondly, visualisation of detected planes is a powerful tool to increase the usability of the application. It does however somewhat reduce the immersion of the experience to have a constant overlay over all surfaces, so some different options were experimented with. For observers it was decided to hide plane visualisation all together as it does not contribute with anything to them. For the master user though, removing it all together would reduce the usability and make it unclear why a model can be placed at one location and not at another. Keeping it on at all times caused a lot of clutter and would greatly reduce the immersion. Some different textures used for visualising was explored and it was also discussed to perhaps have only the edges of a plane be visualised. In the end a satisfying compromise was found where plane visualisation would be visible up until the first model is placed by the user, and continue to be visualised while an object is selected but be hidden while no object is selected. This sort of gave the master user an "Editor mode" and a "Viewing mode" that they can toggle between and reduces the amount of information available to them when it is not needed.

Finally, some basic onboarding instructions were added and as suggested by the AR guidelines by Google, an animation was added to show how the user can move their device to detect surfaces. Also, most buttons in the 2D interface were changed from transparent to be given a background and appear as floating action buttons, as the transparency made them too hard to see and interact with, which was not considered to be worth the generally less intrusive appearance of having transparent backgrounds. The concept and prototype ended up being referred to as AR-Showcase.

The final version of AR-Showcase is presented in section 6.1.2.

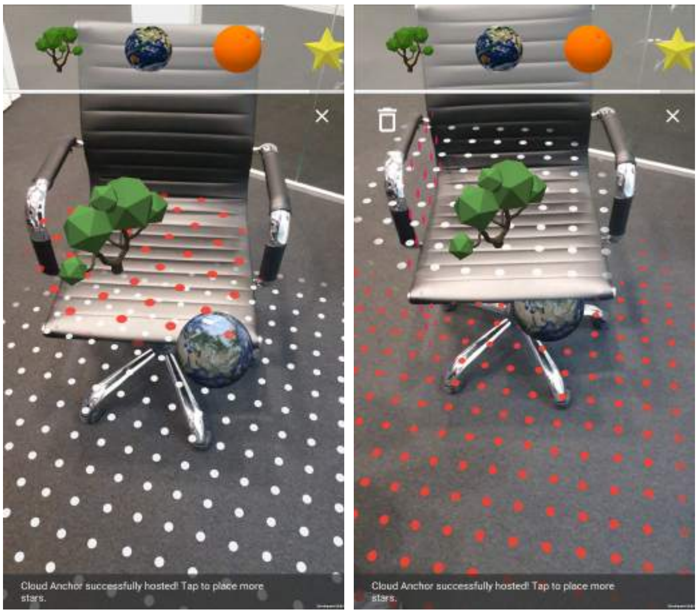

Figure 5.6: The left image shows before occlusion is applied and the right image shows after.

5.2.3 Evaluation

After the prototyping phase it was time for evaluation. It was decided that the tests would start with the users trying out the application and since the prototype is a multi-user application, it was desired to test with several users at once. It was also decided that the test would make use of the think-aloud protocol, to better understand the initial thoughts and expectations of the users. After the users had tested the application for a while there would also be an interview, with all users that were involved at the same time. The reason for doing it with all at the same time was that the users was going to use the application together, and that it was desired to get a discussion going during the interview. In order to get a discussion, it was decided that the interview would be semi-structured. The goal with the user tests was to gather information, thoughts, ideas and insights that would be used as the base for our guidelines.

According to Nielsen [78], five is a good number of users when doing user tests as this allows you to identify most of the issues present in the design. The tests for the prototype were therefore conducted twice, with a total number of five testers. For the first test, three employees at Inceptive and their sister company Uptive were asked to participate. Among these people, one person was the CEO of Inceptive, one was a UX designer and one was a developer working within the area of computer graphics. The second test was carried out on two students from the Interaction Design and Technologies programme at Chalmers, where one had a background in software engineering and the other one had a industrial design background. These five testers were picked since they together have a good mix of AR knowledge and design thinking. Participating in these tests were voluntary and the participants were not given anything for doing the tests. The observations and interviews from these tests can be found in Appendix A. The interview questions were initially formulated in English but since the interviewees were Swedish speaking, the questions were verbally translated during the test and the answers were noted down in Swedish.

The user test were carried out according to plan, the users were given an introduction to the application through an explanation of a scenario in which such an app could be used, in this case where the master user is a teacher and the observers are pupils. Then the users got a few minutes to get acquainted with the application by simply experimenting with it while following the think-aloud protocol. After some minutes the master user was changed, so most of the users got to try out both the role as a master user and as a observer, to get a common ground for discussion later on in the interview. After the users had freely experimented with the app for a while, a specific task was presented to them. The master user was to place a model of Earth, and show to the observers where a specific country or continent was located. This task was presented to observe how the master user and observer would engage with the app when several users needs to see some specific information in AR at the same time. There was not any specific time set for how long the users should try out the application, instead the test continued with the interview when it felt like the users had tried everything out. However, trying the application and conducting the interview took about one hour.

After the tests had been carried out and the data was collected, an affinity diagram was made with all observations from the tests. The observations were clustered together in order to find common problems or good things about SMARE or the prototype. Then, based on the result from the affinity diagram and the answers from the interviews, guidelines were created. There were some observations and answers that came up several times during the user tests and were as a result added to the guidelines.

Since the initial guidelines were mostly grounded in already existing AR design guidelines, they were interesting areas but did not yet take the SMARE aspects into account. After testing and evaluating the prototype one important factor that would influence many of the guidelines was the varying amount of people that an experience could be designed for. Applying this variable to the initial guidelines instantly gave some ideas for new and updated guidelines. An example of this is how large area the experience is designed for. Now you also need to consider that it should fit several users, and the more movement-oriented interactions there are, the more cramped the area will feel as a result of everyone requiring more individual space. This also ties in to the safety aspects considering you can now have several people running about and it is important to keep them all safe during the entire experience and not encouraging any dangerous behaviour. Insights and feedback from the evaluation also pointed towards the fact that being able to scale objects could help in designing experiences for different sized play areas and a varying amount of users.

When the users tried out AR-Showcase there was a lot of communication between them. In these tests there were only two to three users at the same time so it worked well, but the users thought that it would not be as good if there was for instance ten or twenty users in a session. It could be pretty chaotic having that many users talking over each other. So it was decided that a guideline about this should be added, that encouraging verbal communication is probably more successful if there are fewer users playing together. The same reasons were brought up when deciding a guideline for sounds from the application. For smaller number of users sounds can enrich the experience, while if a big number of devices are playing sounds it is possible that it would be annoying to hear.

A really interesting insight was that users gave feedback that they would like the ability to individually rotate and potentially move objects to be able to see them from a direction of their choice, similarly to the initial guideline on accessibility. This does however also have the implication that everyone will see the same object from different angles and potentially at different positions, which would would cause them to no longer be synchronised and as such diminish the experience of object presence and seeing the same object at the same place, in the same manner as everyone else. It is an interesting aspect to consider and whatever you find most important, it is always a trade-off. In a similar vein, another observation was that the users who could not directly engage and interact with the system tended to lose interest faster than the one dictating the experience. As such it could be worth for the designer to consider involving all users in the experience and keeping them engaged.

With SMARE guidelines pretty much being an extension of regular AR guidelines, it also brings additional issues. An example of this which was also experience during the test is that if some technical issues arise while the users are connecting to each other, experiences could be unsynchronised and virtual objects would not appear in the same place. This caused a lot of confusion for the user who was not sure if something had gone wrong and how they could fix it. Usually it was solved by exiting and trying again. Another interesting insight was that depth perception in AR, which is mentioned in regular AR guidelines, had additional implications in SMARE. Since a user sometimes had difficulties determining the depth of an object, users that looked at the same object from different angles would have a similar understanding of where the object was placed. In a similar manner, taking breaks to avoid user fatigue is mentioned in previous guidelines but in SMARE you need to consider how this can be designed around for several users.

5.2.4 Guidelines

The following guidelines are the results from the first iteration of the process and are primarily based on the initial guidelines, insights made during development, and the testing and evaluation of the first prototype. As such, these guidelines are the first iteration of guidelines and following iterations may see guidelines being removed, added or changed.

- Giving everyone the ability to individually manipulate their own instance of an object through actions such as rotation, scaling and translation can increase the accessibility [44], but may diminish the experience of object presence and seeing the same object at the same place, in the same manner as everyone else.

- Sounds can enrich the AR-experience [44], [75], but consider how your experience is making use of it. If many users are in the same room and connected to the session, hearing the sound from other devices could be confusing.
- Consider how many users the experience is intended for and how big area they need. If it is intended to have many users in a smaller area, the experience should probably be designed around requiring less movement from the user as the area will be rather cramped. A smaller amount of users in a larger area allows for more movement without disrupting the experience.
- Be sure to keep all users safe during the entire experience. If the intended area is in a potentially dangerous place, consider giving the users reminders that tells them to keep track of each other and their surroundings and/or give them the ability to define the experience area initially to avoid someone walking into a dangerous spot.
- Giving users the opportunity to scale objects can help fitting the applications' area in both smaller and bigger physical spaces and for a varying amount of users.
- Tell the user if something is going wrong and the experience is unsynchronised, and find an easy way to restart or fix the problem.
- Consider how much verbal communication is required between users and how many users the experience is intended for. If the experience is intended for a few number of users, it can be very beneficial to encourage verbal face-to-face communication [79]. However, having a large amount of users all talking over each other may cause a very chaotic and annoying experience. In this case it might be beneficial to have other ways of interacting that does not require as much talking.
- Perceiving depth of virtual objects can sometimes be difficult and may cause users viewing the same object from different perspectives to perceive it differently. Use techniques such as occlusion and placing shadows beneath virtual objects to give better object presence and help the user perceive depth [44].
- Consider involving all users in the experience. Users that can not interact with the system for longer periods of time can lose interest.
- Consider that users can be fatigued when using AR for a while. Encourage users to move their phone around in order to change the position in which they are holding it [44], [75]. Also, try to avoid forcing users into long play sessions by allowing a user to take breaks without disrupting the experience for others, or make it easy for users to all take a break simultaneously, pausing the experience.

5.3 Second Iteration

This section goes through the ideation, prototyping and evaluation phases of the second concept, and how this led to the second iteration of guidelines.

5.3.1 Ideation

The first step of the second iteration was another ideation session. The goal here was to come up with an idea that was very different to the first prototype, to be able to explore other aspects of the SMARE design space. In contrast to the first prototype, where one user was the master user and had most control and interactions while the others acted mostly as observers, it would be interesting to create an experience where users have similar roles and all interact with the same objects in a collaborative or competitive manner. For this purpose it seemed fitting to make a game.

During brainstorming, a number of different game ideas and genres were brought up with a focus on collaboration and competitiveness. Since the goal was not to create a completely new and unique game concept, but to use the game as a means of further exploring SMARE, already existing game ideas within other areas were also considered. Overcooked [80] was one the games which were heavily considered to be used as a base, as it had many of the dynamics which would be beneficial to testing the prototype. For example, by giving the users more tasks than any one of them can handle by themselves promotes teamwork and communication to successfully complete a level while not limiting some possible interactions to any one user. To create something similar, even though very simplified, was deemed to be too time consuming as it includes many different mechanics and distinctive graphics to work as intended. Other couch co-op games as well as board games were looked at for inspiration.

In the end, it landed at making a racing game. Making a racing game would allow everyone to gather around an area on the floor, or a table to play in manner similar to regular tabletop gaming, but allowing for real-time gameplay by utilising AR. Additionally, everyone control their own car through their mobile device and interact with other users as they try to beat them by either directly colliding with them or by picking up power ups which could allow for other types of direct or indirect interactions with other players. This idea was inspired heavily by the Mario Kart series [81] as well as other similar games. One major difference with the game being developed for this prototype, called AR-Racing, is that it is not being viewed as an over-the-shoulder perspective, similar to Mario Kart, but instead the user can freely look around the racetrack which is placed on the ground or table in front of them.

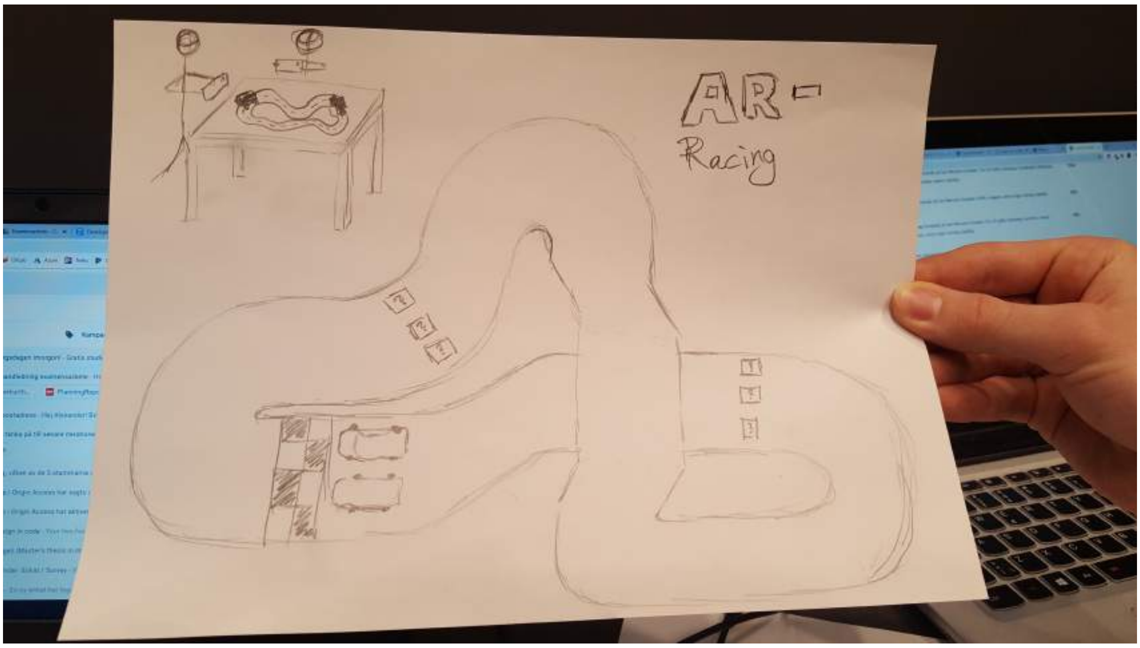

Figure 5.7: Initial concept for AR-Racing.

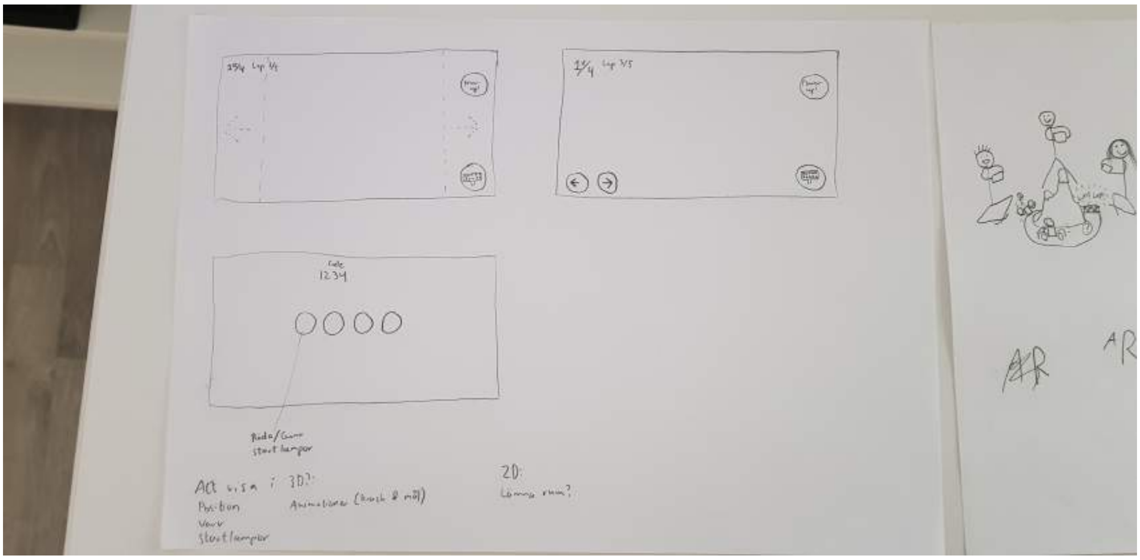

Figure 5.8: Initial sketches of the UI for AR-Racing.

5.3.2 Prototyping

The plan was initially to create a quick physical prototype to evaluate the concept and find any glaring flaws, similarly to the one in the first prototyping session. As always though when it comes to real-time aspects of designs, they are challenging to test and get any valuable feedback from in a low-fidelity prototype such as a paper prototype. With this in mind and the fact that the UI of the design was minimal, it was decided to skip the physical prototyping and instead start with the digital prototype.

The AR-Racing prototype would be very different from AR-Showcase in many ways but the base would still be using ARCore [9], PUN [33] and other back end functionality. Creating a new prototype from scratch would therefore mean redoing a lot of work which has already been done for the first prototype. With this in mind and the fact that a third prototype would also be developed eventually, an effort was made to extract the code base with the underlying technology that could be reused for this and future projects. While this took some time, which was not spent directly on prototyping, it hopefully saved some time in the long run.

To get something up and running as fast as possible, creating a racetrack and a vehicle was the highest priority. Creating a racetrack would include straights, curves and potentially elevations. Without much experience within the area, the Unity Asset Store [82] was browsed to find any potential assets that could simplify the track building. After some browsing, the Dust & Rust Racetrack Pack [83] was found, which included some models for straights, curves and decoration, in a cartoony low-poly style which seemed to work well with the mobile platform it was being developed for. Having some elevated parts of the track would be cool but it was ultimately ruled out as it was not included in the pack and would cause unnecessary complications. The pack came with an example track, which was used through the earlier stages of prototyping.

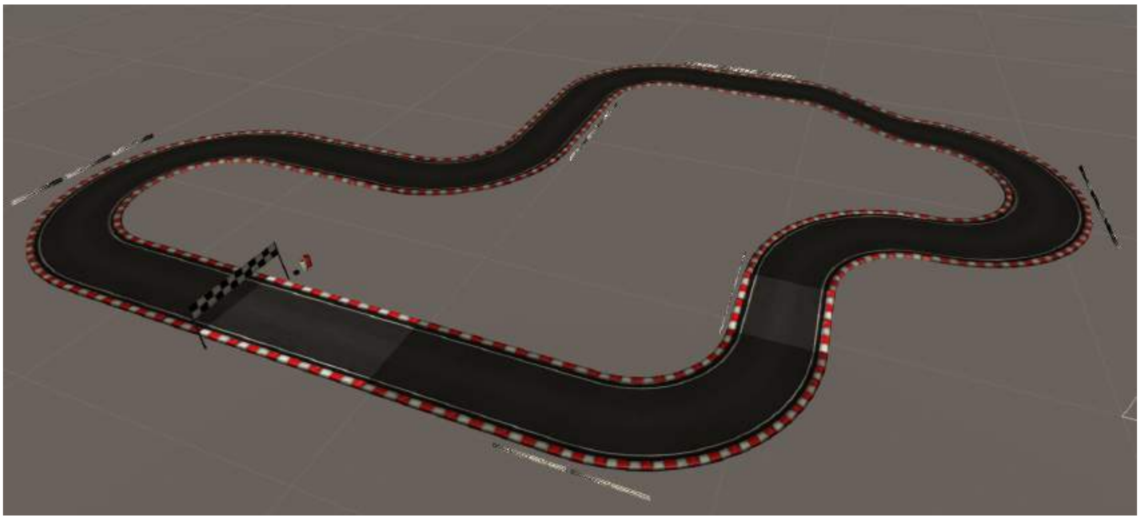

Figure 5.9: First version of the racetrack.

After creating the track, it also needed to be placed in the environment by the person who created the room. This was done by displaying a see-through version of the track which aligned to a tracked plane in the direction the user looked. It could thus be moved around by looking around and it could also be manually rotated and scaled through touch gestures, to accommodate for different play areas. The blue tint and see-through was applied to make the user understand that the track was not actually there yet and about to be placed. By simply pressing the screen the actual track would get placed. Later in the prototyping phase, plane visualisation was disabled for all parts of the experience except for during track placement to help guide the host and give a better understanding of where the track could be placed.

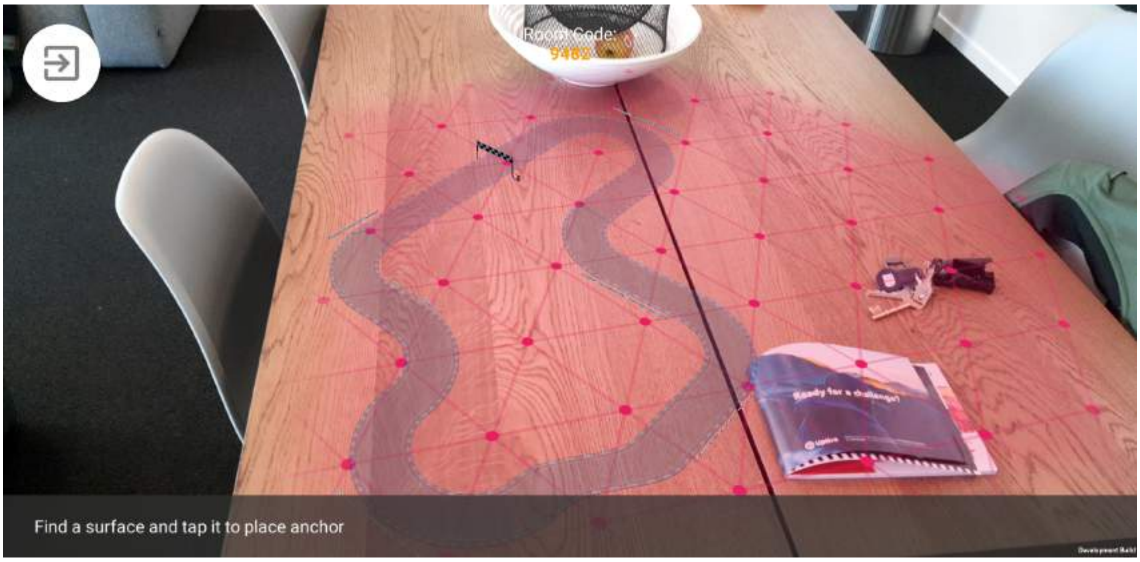

Figure 5.10: Preview of the racetrack before placement.

The vehicle was a bit tricky to implement and caused many problems. At first, a 3D model of a car was found in Unity Asset Store [84]. From the beginning the idea was to simulate the physics as if the car was a big box. To move the car, forces would be applied to this box and it would slide on the ground. By doing this, the physics of the car would be very simplified. It was soon realised that this would give the car a pretty bad hit-box since Unity would treat the car like a box rather than the shape of the 3D model. Since this could make collisions look weird, it was decided to try to go one step further. That was to get rid of the box that covered the entire car, and instead use a custom hit-box for the body of the car. This made the collisions look more realistic but with the downside that it would require more computational power when playing the game. The wheels was not included in the custom hit-box, at first they where represented by two cylinders, one for the front wheels and one for the rear wheels. This representation would result in more realistic collisions in the game, but Unity would still treat the car as an object that would slide on the ground when being pushed, which totally works but it does not look and feel very realistic. To get rid of the sliding behaviour of the car, the cylinders were pretty soon exchanged to Unity's own, more advanced representations of wheels which gave the car much better movement. Instead of the sliding behaviour, the car was now simulated as if it actually had four wheels that it rolled on. Also, instead of applying forces to the entire car to move it, it was now possible to add torque to the wheels in order to make the car accelerate and brake. The wheels also contained functionality for steering the car by turning the wheels. All these things that the wheels provided made the car look and feel much more realistic and it was an huge improvement compared to how it was before.

The wheels had a lot of parameters that could be tuned in order to get the car behave exactly as wanted. Friction was one thing that could be changed and it was hard to find the perfect values. Too low friction would make the car go straight forward when trying to turn, and too high made the car roll over in the curves. In the end a solution to this problem was found, and it was to have a pretty high friction but at the same time lower the centre of mass for the car. That made the car turn without rolling over.

The probably biggest and most time consuming problem during the making of AR-Racing was to get all virtual objects in the right scale. The original size of the 3D model of the car was pretty close to the size of a real car, and the plan was that the racetrack would fit on a table in the end. This did not seem like a big problem and the car was scaled down to the desired size. The problem was that objects which are affected by physics cannot be scaled in Unity without negatively impacting how the object is affected by physics. The gravity looks awful and the car which worked totally fine in normal size was now undriveable. Instead it was now shaking all the time and could not turn. The fact that the game was going to be in AR did not make it easier. In a normal computer game, the environment could have been scaled to fit with the size of the car, but in AR it is not possible to scale the environment since it is taken from the real world. Instead a solution was found, which involved scaling all data that came from the user's device. The result of this is that the device thinks that everything is placed further away from it than it actually is. In AR-Racing the default scale is 100, so if the device is one meter from a surface, it instead thinks that it is 100 meters away from it. This will make the objects appear to be smaller without affecting the physics.

There were many different ideas in the beginning of how the player should control the car. It was difficult to decide how the interface should be designed, so a few non-AR mobile car games were downloaded in order to see which kind of controls they had used. The interfaces had their strengths and weaknesses but there was a game called Beach Buggy Racing [6] whose interface felt very good overall, and it also felt that it would work in AR. Beach Buggy Racing was one of several games where the vehicle is auto-accelerating and the player has to brake themselves. This felt like a good solution since there is limited screen space on a mobile phone, and needing to use one finger to accelerate did not feel that good. Both because the player very often is accelerating in racing games, and when playing on the phone the finger would block screen space for big periods of the session. So auto-accelerating felt like a good solution also in AR.

In Beach Buggy Racing, the player steers their car by tapping the screen. If the left half of the screen is tapped, the vehicle steers to the left and so on. Other games used solutions like arrow buttons or phone tilting in order to steer, but since it was desired to keep the interface as clean as possible in order to make space for the AR camera view, the tapping areas used in Beach Buggy Racing felt better since it did not require more buttons in the interface. Phone tilting also does not require any more components in the interface, but that idea was rejected since there was a fear that it would mess up the tracking of surfaces.

Even though it was desired to have as few buttons in the interface as possible, it was decided that it should contain two brake buttons, one on each side of the screen.

The reason for this was that there are situations where the user wants to brake and steer at the same time, and then the user can brake with one finger and steer with the other regardless if their steering right or left.

Figure 5.11: UI for Beach Buggy Racing [6] and for AR-Racing

When the car and track could finally be tested together, the realisation was that it was very easy to drive off the track, which also meant falling forever since there was nothing to drive on outside of the track as it was implemented then. To improve this issue, a new track was created using the same track pieces but this time increasing the width of them and rearranging them a bit to both make it easier to stay on track and making it bit more exciting compared to the first one.

From the beginning, all the cars were orange. It could clearly not be like that in the end, because it was hard to know which car that was yours. At this stage, the only way to know which car that belonged to who was to look at the starting positions before the race started. The host always got the car with the leading starting spot, the first user that connected to the host's session got the second spot and so on.

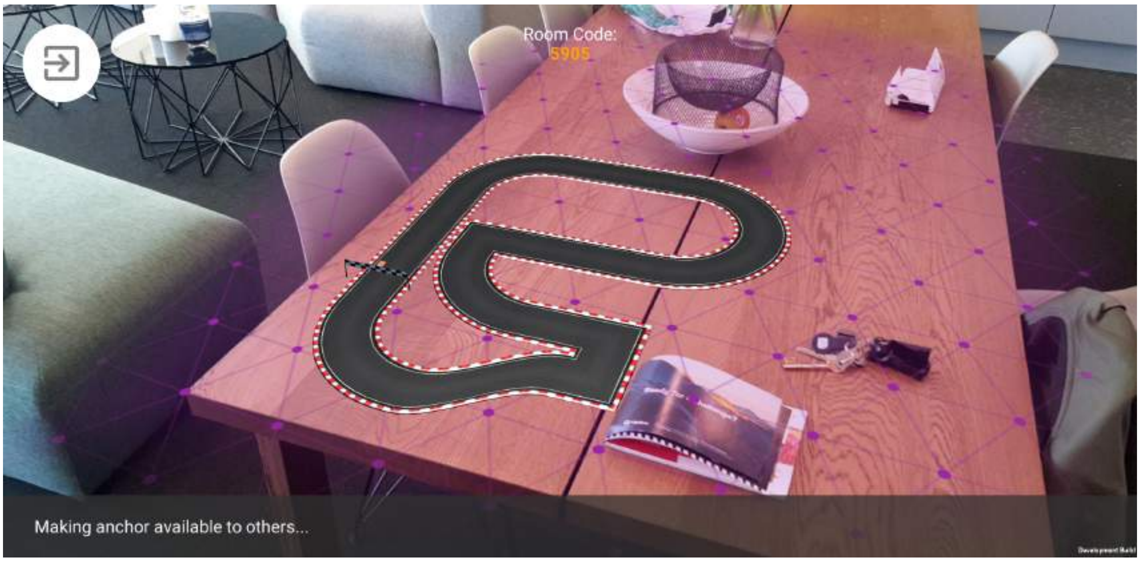

Figure 5.12: Second version of the racetrack.

Therefore the cars were changed and every car got a unique colour. It made it easier to know which car that was yours during the race, but it did not show which colour you got. So if two players connected roughly at the same time, you did not know if the second or third car was yours. Instead you had to see which car that reacted on your interactions. Therefore a circle was added that would be visible around the user's own car before the race started, so the users easily could see which car they were controlling.

The initial concept for AR-Racing included powerups, which users would be able to obtain by driving through an object on the track. These would then be activated to either give some benefit to the player or disadvantage to their opponents. When the implementation of powerups started, it was decided to start with something simple. So the first two powerups would be a personal boost to speed and one that would slow down all other players. Since the cars used realistic physics including torque, velocity and other factors rather than just moving at a set speed made it rather complicated to create powerups that felt good to use. Some parameters was experimented with such as increasing maximum torque but it did not work as intended and while it might increase the top speed of the car it was barely noticed since it still had to be reached by accelerating. It did not feel like the burst of speed which was aimed for. Dissatisfied with the results, that powerup was put on hold while the other one was experimented with. The idea was that perhaps the slow could turn off the motor or brake for everyone else. Similarly to the first powerup though, this did not feel good due to the realistic physics of the car. In addition to the powerups feeling really bad, it was also pretty difficult to both stay on track and use powerups in the current state of the game so it was decided to put powerups to the side and focus on other important aspects of the game that was lacking due to the set backs the issues with scaling had caused.

Collision between cars was something which was changed and debated back and forth throughout the development process. At first, having users be able to collide

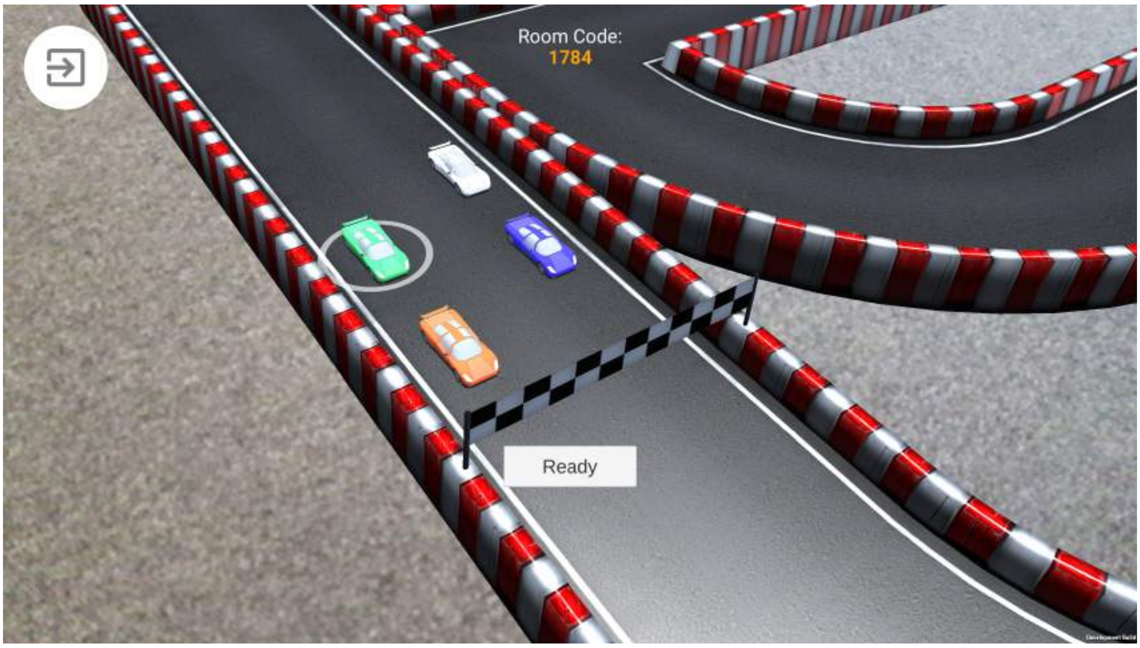

Figure 5.13: Cars with different colours and an extra circle around the user's own car.

with each other was considered an essential part of the concept as it would give the users the ability to directly interact with, and affect the outcome of each other which would cause some interesting dynamics to analyse when testing. When the cars were actually implemented though and could be tested, there was a noticeable delay between where one user had their car and the other users could see their car. This was a result of it being networked multiplayer and the fact that it takes time to send data over the network. As such, collisions between users would be very wonky and what happened to the cars would not be in line with what the users could see. So for the time being, collision was disabled.

Later in the process, the issue with cars being delayed between users appeared again and was deemed detrimental to the experience. As such, an effort was made to research how this could be improved. Eventually, a solution including lag compensation was found. Basically you check the time when the last position of a user was sent, what time it was received, and calculate where the car probably should be by now at the speed it is currently going. This massively improved the previous issue even though it did cause some minor stuttering for networked cars, especially when turning and colliding with walls. This was however deemed a minor issue and as an overall improvement, as now you were able to drive your car through the track while looking at another user's screen, showcasing the improved responsiveness. As a result of this, collision could be enabled once again and it felt really good compared to before.

As mentioned earlier, it was quite easy to drive off the track and this would cause the car to fall forever as there was no collision outside of the track. A few different approaches was discussed to improve this issue. One solution would be to add some ground outside of the track which the user could drive on, or to add lava, water or some other terrain that would cause the user to respawn somewhere on the track. Another solution would be to add invisible walls or props around the track that made it so the car could not drive off the track. At last, a really simple and effective solution was found where it was possible to simply scale the track in the y-axis which would cause the curbs of the track to raise and create a wall around the track. This change made a huge difference as it made the race a bit more relaxed, less frustrating and as a result more fun.

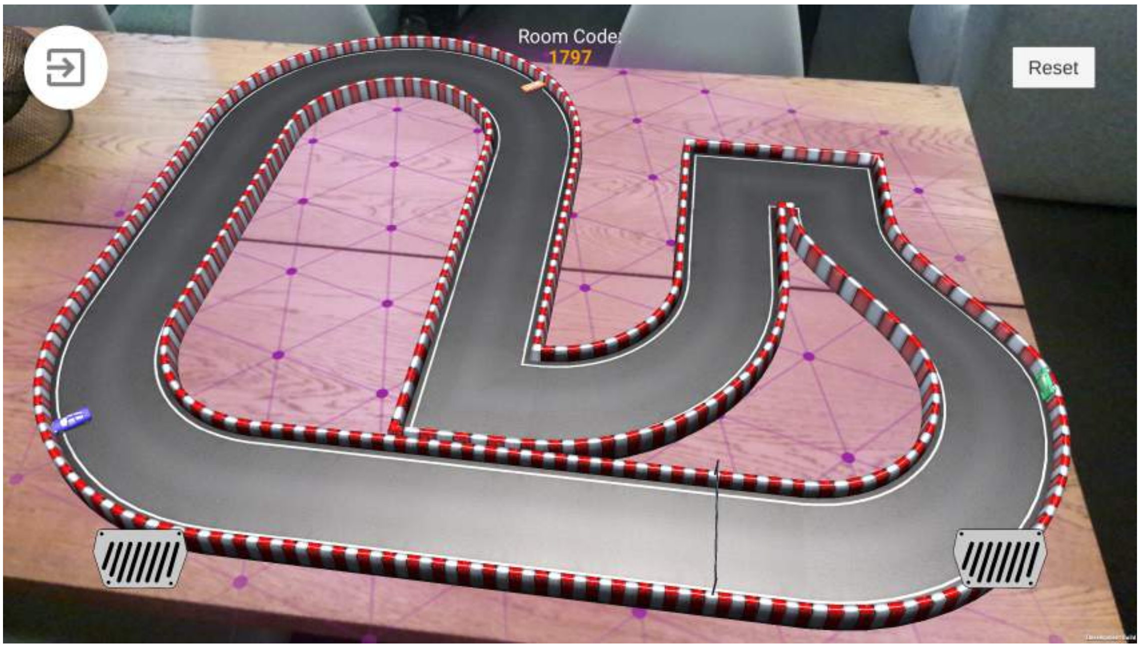

Figure 5.14: Raised curbs acting as walls around the racetrack.

Previous to adding the higher walls to the track, a reset button was added for the host to be able to restart in case everyone drove of the track. This was primarily placed there to help while testing to reduce the amount of times you would have to restart the application. This button was later refashioned into a rematch button which would appear once everyone had finished the race. There was however still the issue of cars occasionally flipping over or getting thrown of the track after a collision. To solve this issue, invisible checkpoints were added at a few spots around the track as well as a respawn button. Pressing the respawn button would return the player to the last checkpoint that they had passed. This also had the added benefit of preventing cheating by backing through the finish line as the user now had to pass through all checkpoints in the correct order.

Since there was a set number of laps that the user needed to complete to win, this also needed to be displayed to the user together with what lap they were currently on. The decision was made to have it as 2D-text on the display as it would potentially be different for each user and therefore seemed fitting for the user's own screen as opposed to being placed in the world. It is still important however to keep the UI clean and not too cluttered as to not reduce the immersion of the AR experience. With this in mind, the lap text would only appear once the player had pressed ready and it would replace the label that otherwise displayed the room code. The room code was not deemed necessary during a race as the application did not support people joining during a race in progress anyways.

At this point in time the race started as soon as everyone pressed the ready button which appeared once you had connected and found the track. The decision was made to add a countdown that would give the players a few seconds to prepare for the start. This countdown was added as a 3D-text in the world that would always stay rotated towards the user's screen, as to explore the effects of this approach to some extent during user testing.

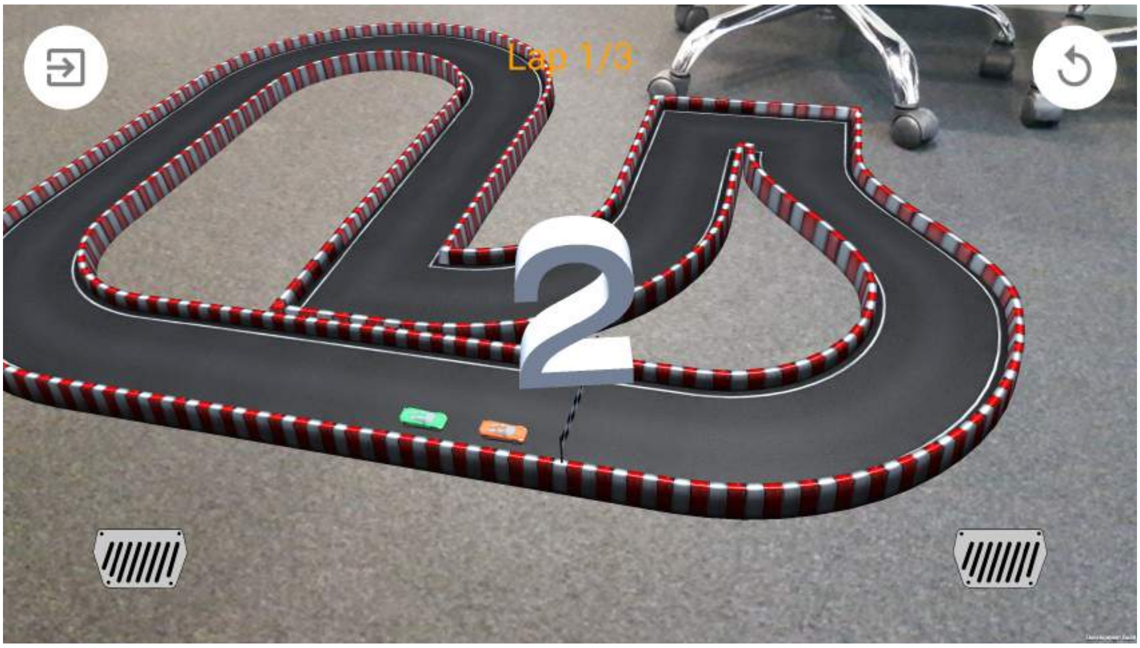

Figure 5.15: A countdown appears once all players are ready to race and lap counter is displayed at the top of the screen

It was desired to have sounds in the application, which was implemented when the more basic parts of the game were finished. At first, an engine sound was added to the car. This sound was simulated to actually come from the cars, so if a car was far away it did not sound much at all, while it would be pretty loud if the car was close to the camera. The thought behind this was to see what these kind of sounds would be like when there are several devices that plays sounds at different volumes depending on their position. What would it be like to see your car on your screen driving to the far side of the racetrack and hear it from the other player's device who is sitting over there? The sound itself was pretty annoying to listen to since it was played more or less all the time, but it was added more because of exploratory reasons than anything else. This was not the only sound that was added though. A cheering sound was added and played every time a player finished a lap. This sound was not placed in the 3D-world as the car sound was, instead this sound was played from the device of the player that finished the lap. By adding this, the players could hear when an opponent finished a lap by listening to whose device the sound came from. When a player finished the race, a slightly longer cheer was played from their phone and if they won the game, an additional fanfare sound was played simultaneously.

When the race was about to start and the countdown was shown, a classic countdown sound was added. A shorter "beep" was played as each number appeared, followed by a longer "beep" when the game started. This sound was played from every player's device at the same time, and there was not any deeper thoughts motivating this sound more than it would be fun and nice to have.

The final version of AR-Racing is presented in section 6.1.3.

5.3.3 Evaluation

When planning for the evaluation of AR-Racing, the plan was to start with letting the test users try out the application and then have a semi-structured group interview. For the evaluation of AR-Showcase, the test subjects were asked to use the think-aloud protocol in order to get a better understanding of how they think while using the application. For the tests of AR-Racing though, it was decided to not use think-aloud. Unlike AR-Showcase, AR-Racing is a game which requires much focus, and trying to explain what you think at the same time as you play the game could be hard. Also, using the think-aloud protocol could interfere with another goal of the tests which was to see how the users communicated with each other, if they for instance trash-talked each other or were commenting on the actions of themselves or an opponent during the races. The semi-structured interviews would be very similar to those of the first iteration. Therefore the interviews from the first iteration were used as a base for the interviews that were going to be held during the user test for AR-Racing. First of all a few questions were removed to try and keep the interview a bit shorter. Some questions were kept just as they were, while others were changed and a few new questions were added to fit the new prototype.

When conducting the tests, the users were first told about the concept and how to drive their car. Once the instructions were explained, one user acted as the host, creating a virtual room and placing the track on a suitable surface while the other user connected to the session. They then did a few races, they were told to do at least two but they could do more if they wanted. While they were playing the game, observations were taken as notes. When the test users felt satisfied and did not want to play anymore, the semi-structured group interview was held.

Similarly to the evaluation of AR-Showcase, the aim was to conduct two user tests on a total of five testers. The first test was carried out on two students from the Interaction Design and Technologies master programme at Chalmers, where one had a background in software engineering and the other one had a industrial design background. These two students had also been part of the AR-Showcase user test. The plan for the second test was to have three testers at the same time. Unfortunately, two of them did not show up due to unknown reasons but another tester was able to show up on short notice and as a result the final test was at least carried out on two people. One of the testers was a developer at Uptive and the other one was a student from the Software Engineering master programme at Chalmers. Participating in these tests were totally voluntary and the participants were not given anything for doing the tests. The observations and interviews from these tests can be found in Appendix B. The interview questions were initially formulated in English but since the interviewees were Swedish speaking, the questions were verbally translated during the test and the answers were noted down in Swedish.

After the tests had been carried out and the data was collected, an affinity diagram was made with all observations from the tests, similarly to the evaluation of AR-Showcase. The observations were clustered together in order to find common problems and good things about SMARE and the prototype. Then, based on the result from the affinity diagram and the answers from the interviews, guidelines were created. There were some observations and answers that came up several times during the user tests which as a result were added to the guidelines.

AR-Racing used sounds in a few different ways which allowed this evaluation to further test the previous guidelines on sounds and expand on that area. One type of audio that was experimented with was having a sound come from one device as feedback on that user's actions. This was observed to be a positive experience in AR-Racing and could as such be something worth considering when designing other similar experiences. Another type of sound, which was experimented with and further expanded on during discussion with the testers, was having 3D-sound that changes in volume depending on the distance to the object. An observation during testing, which was discussed further in the interview, was that this could potentially be used to create a surround sound effect which could enhance the experience in different ways.

While these were some of the ways in which sound could be used to enhance the experience, some limitations were also identified. An example of this is when several people have some constant audio playing. If the audio is not completely synchronised between devices, so that there for example is a delay for some users, it can become rather noisy and annoying rather than enhance the experience. Another example is as the number of users increase, having many different sounds playing all at once from different devices could be very confusing and would be something to take into consideration. As such the guideline from the first iteration that covered the use of sounds was changed and few additional ones were added to further expand on the use of sound in SMARE.

One thing that was added to AR-Racing as an experiment was the race countdown. The fact that the countdown was always facing the users and the rotation of the text therefore not being synced between the devices was not anything that the test users thought was bad. It was more important that the users could easily see the text instead of it being totally synced, since in this application you do not see two screens at the same time so you do not notice that it is not synced. Based on this feedback and the initial guidelines on accessibility, a new guideline was added about always making text in the 3D-world facing the user.

Compared to AR-Showcase, the users that played AR-Racing were much more focused on their own device. They did not look at each others devices or pointing out things in the AR-world as the users in AR-Showcase did. The only interactions between the users outside of the application was talking about what happened in the game. This seemed to be because AR-Racing required the users focus all the time and not letting them take spontaneous breaks, so a guideline was added for that.

5.3.4 Guidelines

In this iteration of the guidelines, one of the previous guidelines has been updated and another five guidelines have been added. This chapter will only cover the incremental changes between the first and second iteration of the guidelines.

5.3.4.1 Updated Guidelines

Below is the updated version of the second guideline from the first iteration of guidelines.

- Sounds can enrich the AR-experience [44], [75], but consider how your experience is making use of it. If the experience is designed for a larger amount of users, consider using fewer, more distinct sounds as hearing the sounds from other devices may cause confusion.

5.3.4.2 Added Guidelines

In addition to the updated guideline above, the following guidelines have been added to the list of guidelines.

- If constant audio is played asynchronously on several devices, it may be annoying or confusing and diminish the experience.
- Consider using sounds as feedback for other users' actions. Playing a sound on one device only can for instance give the users a hint that the user with that device has achieved or completed something.
- Having sounds that originate from virtual objects and increase in volume the closer they are to a device can create a surround sound effect which increases the object presence, gives users a better understanding of depth and helps them in locating objects.
- Visualise information such as text and instructions in a way that it is visible from any angle for all users who may need it [44].
- Experiences which require more focus from the user tend to reduce the amount of face-to-face interaction between users.

5.4 Third Iteration

This section goes through the ideation, prototyping and evaluation phases of the third concept, and how this led to the third iteration of guidelines.

5.4.1 Ideation

The plan for the final iteration had intentionally been left pretty open to explore aspects that had not been covered in earlier iterations. As such, the ideation session started with a discussion to define what would be the goal of the final prototype. It was decided to try to come up with an idea that would apply some of the guidelines previously defined to test them further and investigate some additional areas while also creating a prototype that could be used within a school environment, as this would also have some added value to Mixtive.

With a rough idea of what the prototype should cover, brainstorming was used to come up with a few ideas that would fit the criteria. One early idea was to have the users combine atoms to create different molecules. Another one was to have a laboratory of sorts where users could experiment with different substances and see them react with each other. One idea which was discussed a bit further as it seemed really interesting at first was to give each user an individual instrument which they could play together to create music, in a manner similar to the Rock Band series [85]. It would however probably require a lot of work to make it work well as you want to have several different instruments and if the sound is not synced correctly between users it would sound weird.

The concept which was decided to continue working on was a 3D-puzzle of sorts. The idea was that a skeleton of the human body would be presented to the users and then fall apart, leaving the users with a bunch of bones which they need to place in the correct spot to rebuild the skeleton. As such, this would be a collaborative experience where users are able to pick up and interact with the same objects. It was also discussed to potentially add some elements to promote collaboration by giving hints to one user where the other one can place their piece. This also had the benefit of being potentially interesting in a school environment, which was one of the sought after criteria. This concept came to be called LumbAR.

5.4.2 Prototyping

As the prototyping phase started, the code base which had been extracted for AR-Racing could also be used as a starting point for this prototype. This allowed the prototyping of the more important parts to start right away instead of spending time implementing the same things which had been implemented for the other prototypes. In addition to this code base, the placement of the race track in AR-racing would work pretty similar to how a skeleton would be placed so this code was also possible to be extracted and reused. Towards the end of the project the same code used for plane visualisation was also extracted from AR-Racing and applied to LumbAR as it would work in the same manner.

From the beginning, the plan was that the third iteration would be four weeks long, just as the other two iterations. After a few days of the iteration though, it was realised that the report of the project needed to be done earlier than initially thought. To get some time to finish the report, it was decided that the third iteration would be shortened by one week. Therefore, some of the features that was planned to be in the prototype was left out.

The 3D-model of the skeleton was taken from a part of AR School [17] which teaches the user about anatomy. The model fitted well into the needs of LumbAR since it could be disassembled so every bone was its own piece. From the beginning it also included other parts of the body like muscles, nerve system and organs, but everything except the skeleton was removed from the model as it was not needed of this purpose.

The initial idea was that when the game started, the skeleton would break and every bone would fall to the ground. The skeleton that was imported to the project was a static 3D-model, without any code or physical forces applied to it. Therefore gravity was added to every single bone. Collision detection was also added to the bones so they would collide instead of falling through each other or the floor. This gave the desired behaviour to the bones, but at this time the problems started to occur. Having over 200 bones that are colliding with each other requires a lot of computational power, which caused the frame rate of the device to drop significantly when looking at the bones, both when they were falling and laying on the ground. To solve this problem, collision between bones was ignored and instead the bones only collided with the floor. This solution made the bones go into each other while laying on the ground, but rather that than heavy drops in frame rate.

In order to add collision detection to the floor, there were a few different solutions. One was that the bones would collide with the real world surfaces that the device had tracked. This did not feel like a very reliable solution, since the plane tracking is not completely correct all the time. Instead, an invisible plane was added to the skeleton model, so the skeleton always has a floor underneath it. This was judged as the better solution, even though it was not perfect. The skeleton is quite big and therefore it was considered that in the typical scenario the skeleton would be placed on the floor, in which case this solution usually works well. If the user decides to place it on another surface like a table though, the invisible plane could be bigger than the table which can make bones appear to fly if they are outside of the table area.

Pretty early in the third iteration it was realised that the user probably should not be forced to place every little bone in the body to complete the puzzle. For instance every finger contains three to four small bones, some of them so small that they were hard to find in AR, and in total there are over 200 bones in the human body. At this time it was decided that some bones had to be grouped together, but there was uncertainty about how big those groups would be. Some users may want more pieces because they think the puzzle is fun or want to learn, while others may want an easier puzzle. Therefore the initial idea was to let the user choose between a few difficulties, where in the easiest one the user could place an entire leg at once, while at the hardest difficulty the user would more or less have to place every little bone. However, as the iteration time was shortened it was decided to only implement one difficulty. The difficulty chosen was the one that would have been the medium difficulty.

Figure 5.16: To the left, the skeleton parts when some bones are being grouped together. To the right, the skeleton without groups.

In this prototype, users needed the ability to move objects in any direction, including vertically, in contrast to the previous iterations where objects would always be attached to a surface. This presented an opportunity to try out a new type of interaction. To pick up and place an object, the user would need to look at an object and hold down anywhere on the screen to grab it. It would then stay at the same relative position to the device which would allow the user to freely move the object as they move their device. Once they release their finger from the screen, the object has physics applied to it once again and fall to the ground or it gets attached to the skeleton if placed in the correct spot. To give feedback to the users that an object could be picked up, a piece of the skeleton would be highlighted as the user looked at it with the centre of the screen. In addition, a small white dot was added to the centre of the screen to make aiming easier and more obvious.

Once the grabbing and placing of bones were implemented, it did not take long until a big issue was identified. When picking up an object and attempting to place it at the skeleton, it would often just fall to the ground due to being in the wrong spot. The reason for this was that it was very difficult to get an understanding of how far away the object was that you were currently holding. Many times you would seem to have it aligned horizontally and vertically but it would either be too far away or too close. In an attempt to improve the depth perception, objects were made to cast a shadow on the surface below them, as suggested by one of the AR guidelines from the first iteration. While this made a slight improvement in some cases, more specifically when placing bones closer to the ground, it did not have too big of an impact when placing most of the bones as the user normally would not see the

Figure 5.17: As the user points the white dot in the centre of the screen towards a bone, it will be highlighted to indicate it can be picked up.

ground while placing a bone on the higher parts of the skeleton. In addition to this, the margin which the user had to put the bone within for it to attach was increased in the depth axis from where the camera looked. This was an improvement which caused less frustration but was not a great improvement to the underlying problem. Later in the iteration a solution was implemented which would place held objects at the correct depth when looking at some part of the skeleton. While this did not directly improve the depth perception of the object, it made depth something that the user would not need to take into account when placing pieces of the skeleton as it would adjust automatically instead.

In the previous iterations, an object which could be manipulated always belonged to just one user which made it rather simple to handle over the network as the owner of an object would just continuously tell everyone else the current state of the object and they would listen to it. In this case however, every user should be able to pick up and interact with every bone. This type of network handling would be a bit more complex and meant that the previous implementation would not work entirely. To solve this issue, two different approaches were identified. One way would be to let the person who picks up the object send its new state to the owner of the object, who then sends it to everyone else. The other way would be to transfer ownership of the object to the person who picks it up, notifying everyone else that they should now listen to that user for updated states. The decision was made to implement the second approach as it seemed like the more common approach and would seemingly lead to less data needing to be sent over the network and also reduce delay between users.

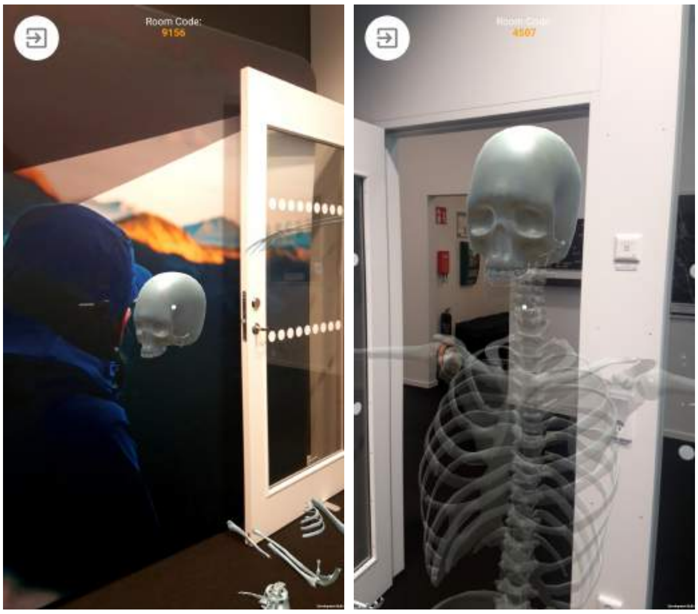

Figure 5.18: The image to the left shows the skull being held in the air by the user. The image to the right shows the skull after being place at the correct spot.

Figure 5.19: Hints for the other user is highlighted in blue.

As one of the aims of the prototype was to create a collaborative experience, there needed to be something to promote collaboration between users and hopefully avoid them silently solving the puzzle individually. Combining this with a way for the users to get hints when they got stuck gave the idea of one user giving hints to the other user. The way this was implemented was that a user would see the object held by the other user highlighted in blue, as well as the piece of the skeleton where it fits also highlighted in blue. This way you could see the solution for the other person but not for yourself, hopefully making people communicate with each other to help when they can not figure something out themselves or to just speed up the process. To avoid confusion, objects are highlighted in different colours depending on if it is you or the other user that is interacting with it.

During the iteration it was realised that bones could be lost, either if they are placed outside of the invisible floor plane, or dropped while being held under the plane. Losing bones under the plane was something that actually happened pretty often. To solve the problem, a big transparent box was added around the entire plane and play area, and as soon as a piece got out of the box, it was considered lost. When a bone was lost, it was re-positioned back to the centre of the plane.

Since the beginning of this prototype, the user have placed a skeleton and when the game starts all the bones just falls from where they are to the ground. That solution works, but the bones does always land on the same spaces. Things like the arms still looks like arms, just that they have been moved from the right position to the floor. This made it easy to know where those bones should be placed. Also the feet were in the same space since they started on the ground, and the user could just tap them and they went into the right space. Therefore it was decided that the bones should be scrambled in some way, but still keep the animation of bones falling down to the ground. The solution was to give every bone a random position at a fixed height and let them fall from there. This scrambled the pieces and spread them out on a bigger area than before.

Figure 5.20: To the left, the bones have just fallen from the skeleton to the floor. To the right, the bones have fallen from random positions.

In the end of the iteration during the time that was left, an effort to implement restart functionality was made. The thought was that when the puzzle is finished, the host could press a button that restarted the game without having to host a new room or place a new skeleton. Instead the same puzzle would appear at the exact same space as before. Unfortunately, due to technical issues and lack of time this was not included in the prototype.

The final version of LumbAR is presented in section 6.1.4.

5.4.3 Evaluation

In the planning phase for the user tests of LumbAR, it was decided to go on with the same type of tests as for the two earlier prototypes; first testing the application and then afterwards having a semi-structured group interview. When trying the application, the participants were asked to use the think-aloud protocol. Compared to AR-Racing, which tests were not conducted with think-aloud, LumbAR felt like a less focus-requiring application where the users could complete the task at their own speed. Therefore, the user would have time to say what they think during the tests and hence it was decided that think-aloud would be used. The plan was to let the users complete the puzzle once, as it would most likely give them enough time to get a good understanding of the application. The plan was to let the users try out the application in pairs.

When preparing the interview, it was decided that it should be a semi-structured group interview just as in the previous iterations. It felt like it worked well in the previous tests, giving good discussions and valuable feedback. Therefore it was decided to do one also for LumbAR. A few of the questions prepared was re-used from the earlier iterations, but the most of them were new for this test. The interview was also a little bit shorter than the others, since the previous ones was a little too long and sometimes a bit repetitive.

During the tests, the users were consciously given very little information. The ones who needed help with setting up the room and connecting the devices to each other got help with that, but apart from that the users were only told that the application contained a puzzle that they should try solve. In that way it was possible to see how intuitive the application was and if the users ran into any kind of problems and got stuck, they were free to ask questions. Most of the users figured out quite fast what the goal was and how they would reach it. They also figured out how features like the hint system worked without any help, although they were a little bit confused until they realised what it actually was. In all tests the users managed to complete the puzzle without any bigger problems.

As with the tests in the earlier iterations, the goal was to test LumbAR on five users, but since this test were done in pairs, three tests were conducted with a total of six users. The first test was carried out on two students from the Interaction Design and Technologies master programme at Chalmers, where one had a background in software engineering and the other one had a industrial design background. These two students had also been part of the tests of the two other prototypes. Test number two was conducted on two persons who also participated in a user test for AR-Racing. One of them was a developer at Uptive and the other one was a student from the Software Engineering master programme at Chalmers. The participants from the third test had not tried any of the previous prototypes. Both of them were students at the Interaction Design and Technologies master programme at Chalmers, one with a background in software engineering and one with a background in industrial design. Similar to previous tests, participating in these tests were totally voluntary and the participants were not given anything for doing the tests. The observations and interviews from these tests can be found in Appendix C. The interview questions were initially formulated in English but since the interviewees were Swedish speaking, the questions were verbally translated during the test and the answers were noted down in Swedish.

After the tests had been carried out and the data was collected, an affinity diagram was made with all observations from the tests, similarly to the evaluation of the previous two prototypes. The observations were clustered together in order to find common problems and good things about SMARE and the prototype. Then, based on the result from the affinity diagram and the answers from the interviews, guidelines were created. There were some observations and answers that came up several times during the user tests which as a result were added to the guidelines.

The difficulties with depth perception and the implications it had on SMARE were identified in the first iteration. In this iteration there was an opportunity to reiterate on that issue as depth perception was a big part of the initial prototype for LumbAR. The approach of designing the placement in a way that the user did not have to rely on the depth of the object seemed to work well as the feedback from the tests was that most users did not have an issue with the depth perception when placing objects. As such, an extension could be made to the previous guideline on depth perception to also cover such an approach.

Observations made during this and previous iterations has shown that having poor onboarding increases the risk of the users setting up the experience incorrectly and getting synchronisation issues as a result. While the exact instructions you provide would be different depending on which platform and technology you use, the general implications of having a good onboarding would most likely be beneficial no matter the platform or technology.

The difference in social interaction as a result of having asymmetric information in the form of hints for the other user, was something which was both observed during the test and reflected upon by the users during the interview. It was clear that there was very little interaction between users before they needed to or had discovered that they could give each other hints. Once this feature became apparent and the more difficult pieces remained, users started to communicate with each other continuously to help complete the puzzle. While this is in no way unique to SMARE, it provides an opportunity which designers within the area may wish to take advantage of.

One thing which was discussed during prototyping was whether or not users should be able to grab bones which are currently being held by another user. In this prototype the decision was made to not let users do so as it could be frustrating and make the user feel like they are not in control when holding an object. Doing the opposite could however be interesting in some cases depending on which behaviour you want to create between users and if it is an important part of the experience.

5.4.4 Guidelines

In this iteration of the guidelines, one of the previous guidelines has been updated and another three guidelines have been added. This chapter will only cover the incremental changes between the second and third iteration of the guidelines.

5.4.4.1 Updated Guidelines

Below is the updated version of the eighth guideline from the first iteration of guidelines.

- Perceiving depth of virtual objects can sometimes be difficult and may cause users viewing the same object from different perspectives to perceive it differently. Use techniques such as occlusion and placing shadows beneath virtual objects to give better object presence and help the user perceive depth [44]. In cases where it is still difficult to perceive the depth of an object, consider designing the experience in a way that the users do not have to rely on the depth of an object to fully engage in the experience.

5.4.4.2 Added Guidelines

In addition to the updated guideline above, the following guidelines have been added to the list of guidelines.

- Provide clear instructions to the users on how they should set up and join the experience. Setting up the experience in the wrong way can cause synchronisation problems later on.
- To encourage collaboration and social interaction between users, consider giving users different tasks or asymmetric information in a way that they need to rely on each other in order to be effective or succeed.
- Think about whether or not several users should be able to interact with the same object at the same time. If several users try to modify an object in the same way at the same time, it may conflict with other users and cause them to not feel in control. On the other hand it could be used to create interesting interactions and behaviours between users which could be an important part of the experience.

6

Results

In the following section the results of the thesis are presented. The results are the final set of guidelines and the three prototypes which were developed in order to explore the domain.

6.1 Prototypes

This section describes the final version of the three prototypes AR-Showcase, AR-Racing and LumbAR. All prototypes have some functionality that they inherit from the SMARE code base, which is also explained further in this section.

6.1.1 SMARE Code Base

There are some parts of each prototype that are the same, which are part of the common code base that the prototypes are built upon. The main menu of each application is one of those parts. It is a very simple menu which consists of two buttons, an input field and a blue background, as can be seen in figure 6.1. The top button is *Create Room*, which creates a new room, generates a random 4-digit room code and starts the experience. The other button at the bottom is *Join Room*, which attempts to join the room with the room code currently in the input field just above the button. If a room with that code is available, it will connect you with the other users in that room and start the experience.

Once the user has either created or joined a room the AR view will appear, as can be seen in figure 6.2. At the top of the screen is a label with the text *Room Code:* and the generated room code right below it. This code is then shared to other users to join the same room as described above. In the top left corner is a button to leave the room, returning the user to the main menu. If the host attempts to leave the room, a confirm dialog will appear to make sure that the room is not closed down by accident, interrupting the experience for everyone. This dialog does however not appear for other users. At the bottom of the screen is a label on a dark background which helps with the set up of the experience by providing instructions on what to do. This label is updated as each step is completed and is different depending on if you are the host or not, as the setup differ between roles. The SMARE code base also includes some of the basic ARCore [9] functionality for setting up an experience so it will track and visualise planes, visualise feature points as black dots as well as host and resolve anchors.

Figure 6.1: The main menu is used to connect users to the same experience.

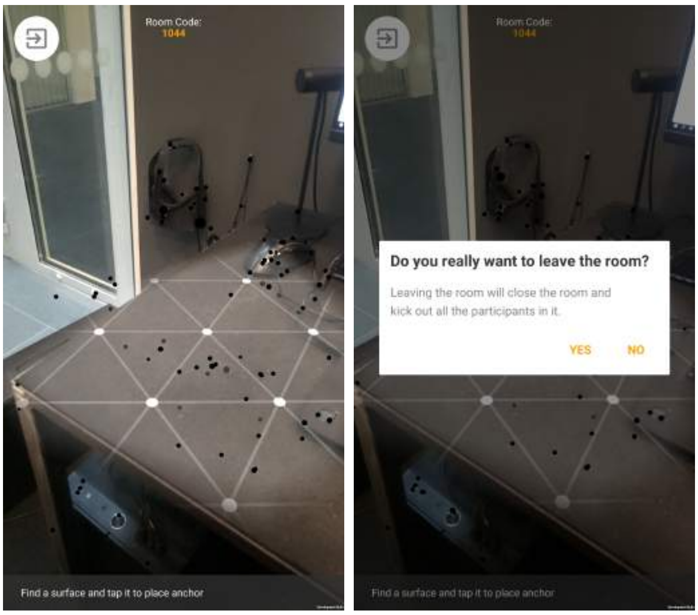

Figure 6.2: The left image shows the AR view of the SMARE code base. The right image shows the confirm dialog for leaving the room as a host.

6.1.2 AR-Showcase

AR-Showcase is a tool for presenting 3D-models in AR, which can be viewed by others through their own device. AR-Showcase uses the SMARE code base as the other applications (see section 6.1.1), meaning that the main menu and functionality for connecting users with each other is the same.

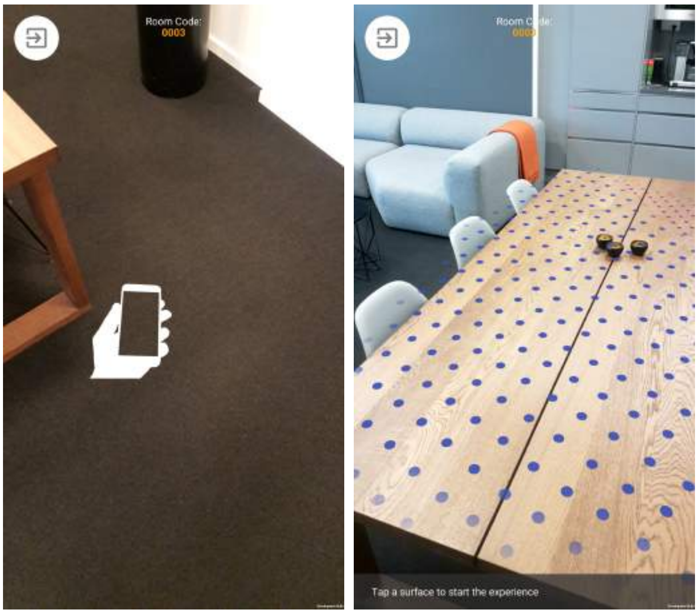

Figure 6.3: The animation with the phone stops when a surface has been detected.

When the host has created a room, the first thing they will see is an animation of a phone in a hand moving around in a circle, as seen in figure 6.3. This hints at the user to move their phone in a similar fashion to detect surfaces. Once a surface has been detected, which is visualised by coloured dots, the animation goes away and instead messages appears at the bottom of the screen that guides the user step by step through setting up the experience. Tapping on a surface will place a compass and start hosting an anchor at that position, as can be seen in figure 6.4. As the compass is placed, a button with a plus sign will appear in the top right corner only for the host. When the anchor is successfully hosted, the message at the bottom will disappear and other users in the same room can start resolving the anchor by looking at the location where the compass is placed. As soon as they see the compass, they should be synchronised with the other users. The plane visualisation is only visible to the host.

If you press the top right button, a horizontally scrollable panel will appear at the top of the screen with thumbnails of the the available models that can be placed. The panel can be closed by pressing the overlapping "X"-button. Models are placed by dragging them from the top panel onto a surface. As a model is placed it will become selected, which is indicated by a white circle on the ground beneath it, as can be seen in figure 6.5. The host can interact with the models, while the other users can only observe and will not see that a model is selected. Only one model can

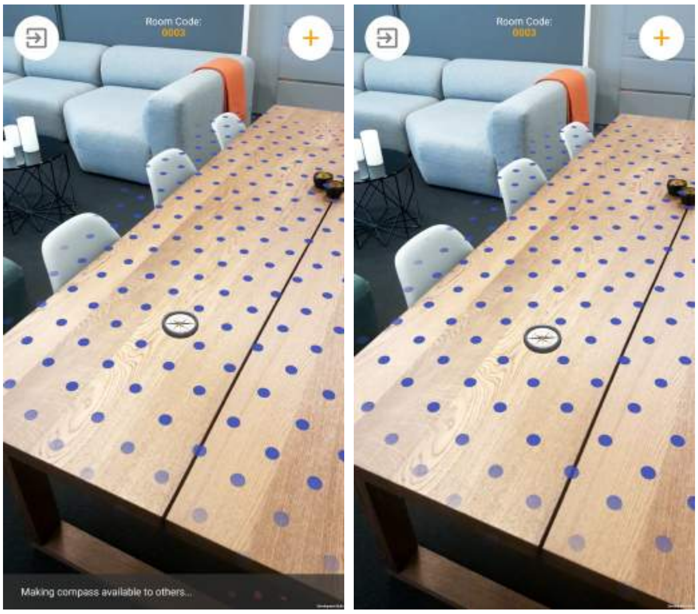

Figure 6.4: Once the experience is available to other users the message at the bottom will disappear.

be selected at any time and as long as one is, another button will be visible in the top left corner with a trash bin icon. By either clicking this button while a model is selected or by dragging the model to it, the model will be deleted. When dragging an item to it and holding your finger over it, it will turn red and the trash bin will open, indicating that you can release your finger to delete the model.

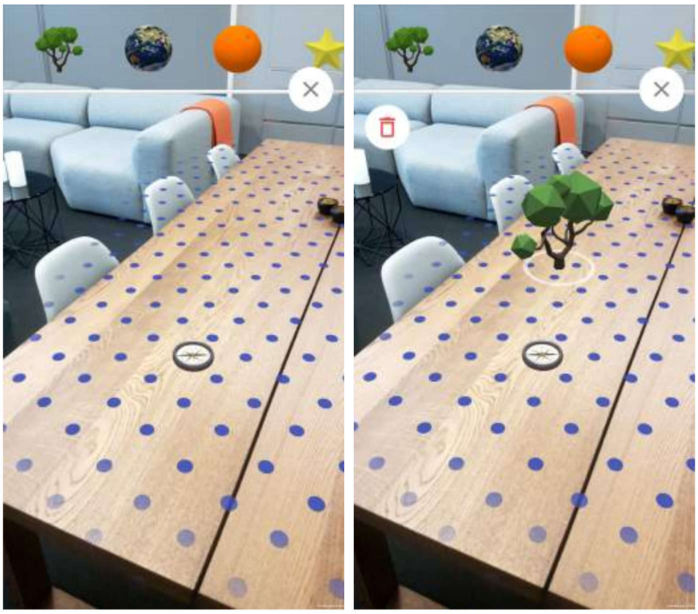

Figure 6.5: Models can be dragged from the top panel out onto a surface.

If the top panel is closed down while a model is selected, the delete button will move up to the top. The delete button will always be visible as a model is selected, otherwise the leave room button will take its place, as can be seen in figure 6.6. A model can be deselected by tapping at an empty area or on another model, as that model will be selected instead. The plane visualisation will also only be visible as long as an object is selected and hidden otherwise.

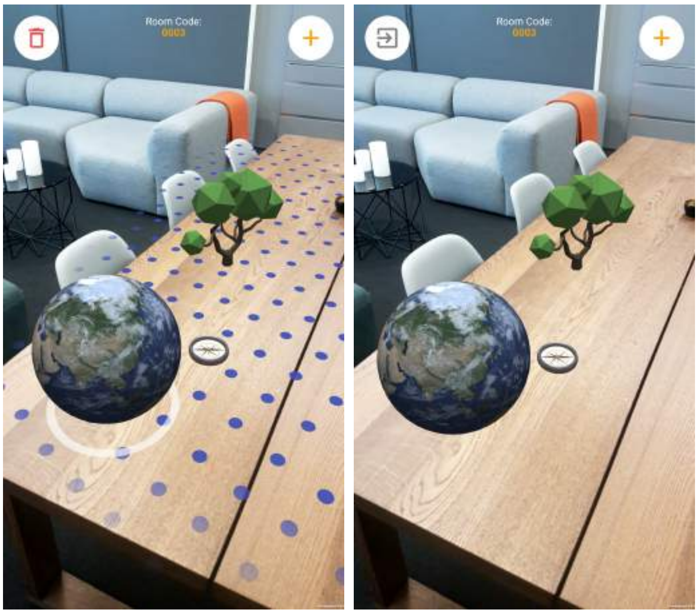

Figure 6.6: Plane visualisation is only visible as an object is selected.

There are a couple of gestures the host can do to manipulate a selected model. First of all they can move it by simply dragging it to a point on a surface. Models can also be resized by doing a pinch gesture (see figure 6.7). They can also be rotated around the y-axis by either doing a horizontal pan gesture or by using two fingers to do a circle motion. As an object is manipulated the changes will be synchronised with other users.

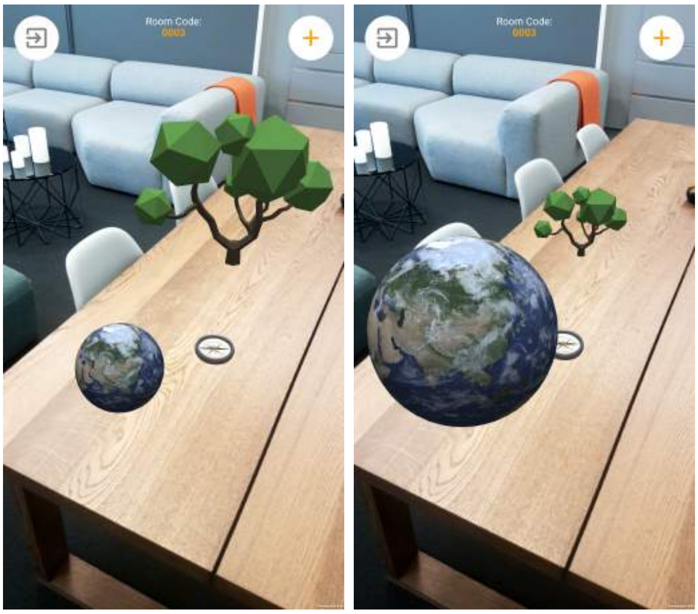

Figure 6.7: The host can be move, rotate and scale models.

As illustrated in figure 6.8, detected planes will occlude virtual objects, making it possible to partially or completely hide models behind real objects in the environment.

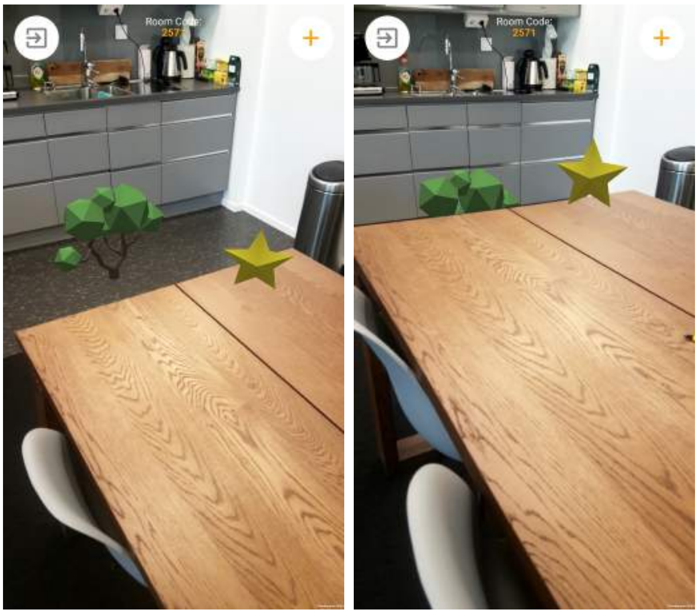

Figure 6.8: Detected planes occlude virtual objects.

6.1.3 AR-Racing

AR-Racing is a car racing game for two to four players. The goal is, like in most other racing games, to be faster than your opponents and be the first one to cross the finish line.

The first thing that the players see when starting the application is the main menu. Since AR-Racing is built from the SMARE code base (see section 6.1.1), the menu looks the same and works in the same way as in the code base. However, the entire application is locked in landscape mode, even the main menu, which is the only difference between the menu in the SMARE code base and the one in AR-Racing. The menu helps the players connect to the same experience and that is also done in the same way as in the code base, one person creates a room and the others connect with a room code.

When the player is connected to a room that they either have created or joined, they will be sent to the AR-scene in which the entire game will unfold. Here, the host will have to place the racetrack. This is done by first moving their device around so the application can track the different surfaces in the physical room. Then, when a surface is found, the host can look at it and a half transparent, preview version of the track will be shown for the host only. It can be moved by looking around, resized by doing pinch gestures or rotated by either swiping horizontally or a rotating two-finger gesture. When the host is satisfied with the placement of the preview track, they can tap the screen and it will be exchanged for the real racetrack which cannot be modified. The host's device will share the state of the racetrack to the other players. First when that is completely done, the rest of the players can resolve the racetrack by looking at its position from the same angle as the host, and when the players' devices find the right position, the track will be shown on their devices as well and they are ready to race. The racetrack contains a mix of sharp and wide turns, long and short straights. It also has high walls that helps the players to not fall off the track.

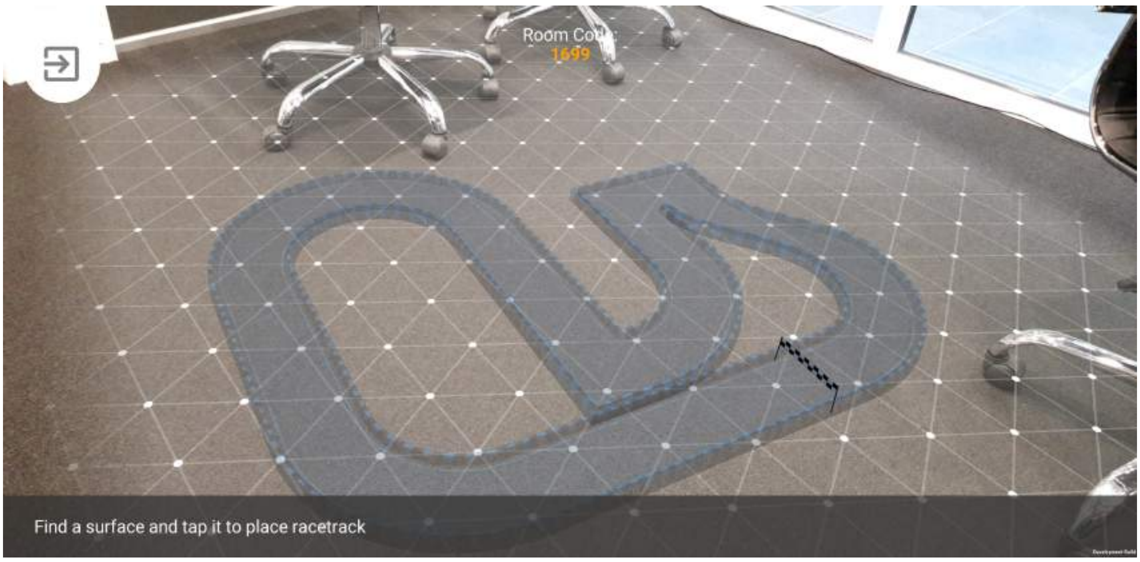

Figure 6.9: The racetrack can be scaled and rotated during placement.

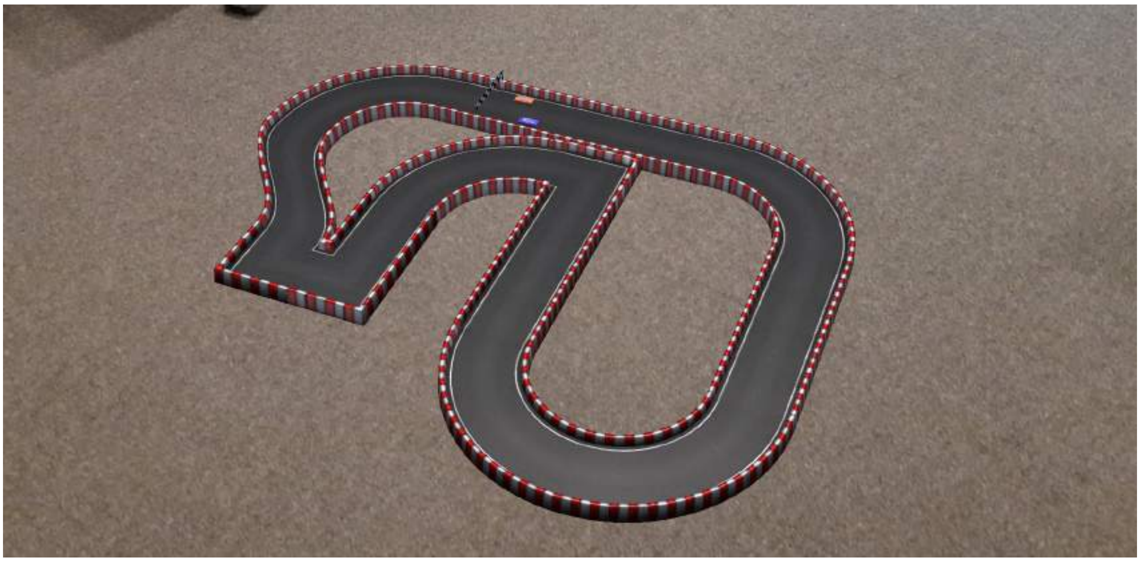

Figure 6.10: The placed racetrack.

Every player gets their own car with a unique colour when they either host or resolve the racetrack. To show the player which car belongs to them, a white circle is shown around their car before the race starts. Every connected player also gets a "Ready"-button in the middle of the screen, which the player presses when they are ready to race. When all connected players have pressed the button, the system consider that as all players that wants to play have connected and are ready to go, and the game will start. A big 3D text will start counting down from three, together with some sound effects. When the countdown reaches zero, the cars start to accelerate and the race is on.

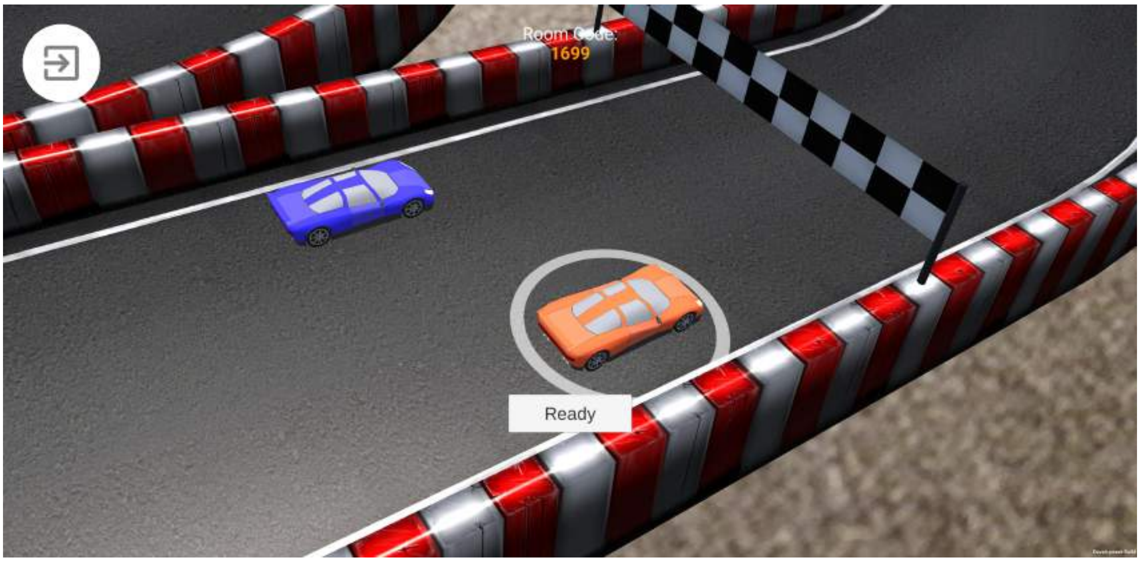

Figure 6.11: The cars standing in their starting positions, the orange car with a circle around indicating that it is your car.

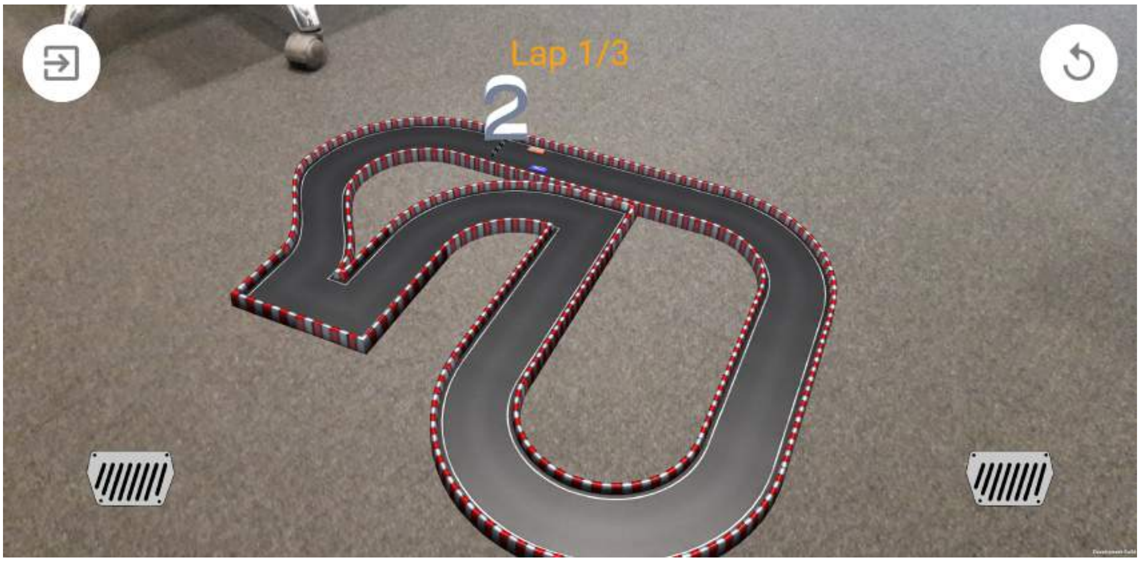

Figure 6.12: The countdown right before a race starts.

The players have a few different interactions they can do to control their car. They can tap on the screen to steer, if they press at the left half of the screen the car will turn left and if they press the right half it will turn right. The car is designed so it will always auto-accelerate as long as the player is not braking. They can brake by pressing one of the two brake buttons that are located in the lower corners of the screen. If the car is standing still, either because the player has drove into a wall or used the brakes long enough, braking will instead make the car reverse. The player also has the possibility to respawn the car if it has for instance went over the walls and out of the track after a collision. This is done by pressing the button in the top right corner, at which point the car will respawn at the last invisible checkpoint the car has passed. In the interface there is also a button in the top left corner, which is received from the SMARE code base and makes the player leave the game and go back to the main menu.

During the race, the car will make an engine sound while driving around. This sound will be louder on devices that are close to the car, but devices that are sufficiently far away will not play any sound at all for that car. This makes it possible for the users to hear roughly where the car is located.

At the top of the screen, there is also a lap counter which shows which lap the player currently is on and how many laps that they should do. Every time a player completes a lap, a cheering sound is played from their device, giving the other players feedback about what have happened. When completing the last lap a longer cheering sound will be played. If the player wins the race, an additional fanfare will be played too, and some fireworks will be fired from the finish line. When a player finishes the race, a short text will appear on the screen displaying which position they got in the race. For the host, a restart button will also appear as soon as all players have finished. Tapping that button will place all the cars at their starting positions again, and all the players will have to press the ready button before the new race starts.

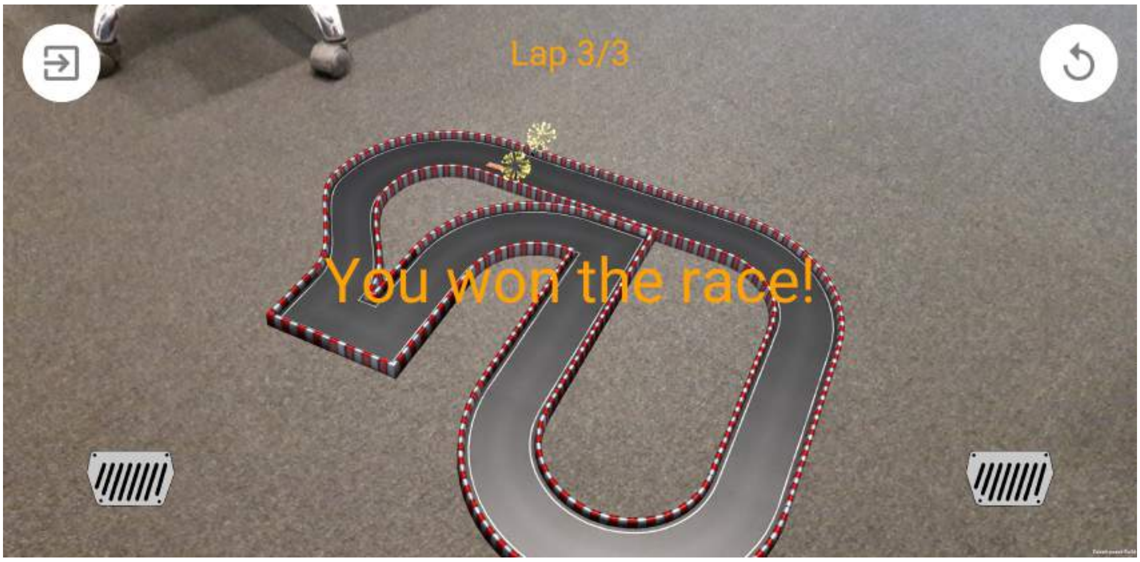

Figure 6.13: Fireworks that are showed when the winner finishes.

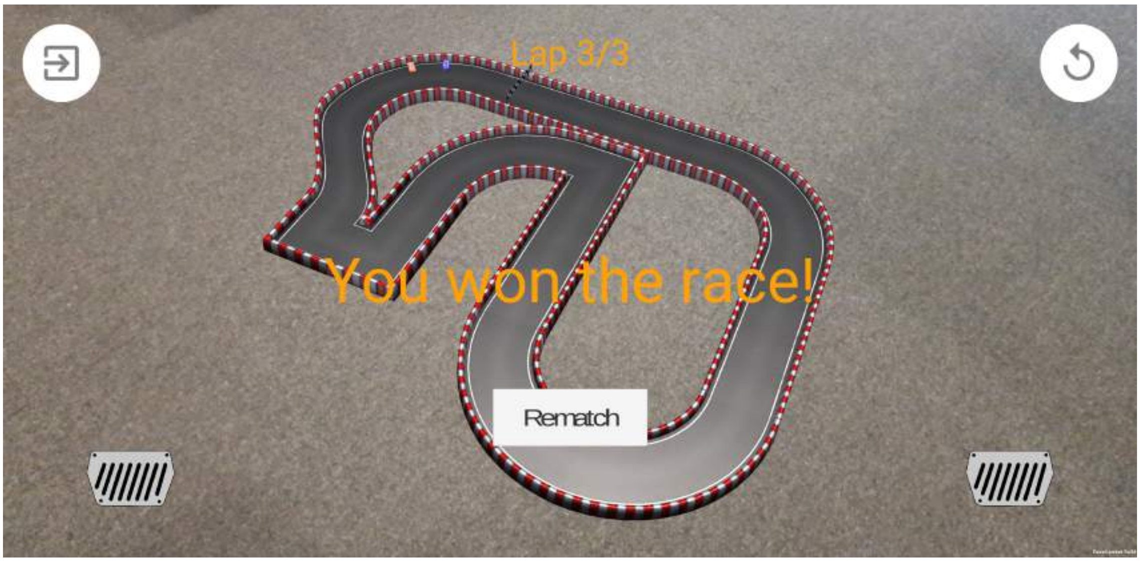

Figure 6.14: The rematch button which is showed for the host when all players have finished the race.

6.1.4 LumbAR

In LumbAR, up to two users work together to rebuild the human skeleton in AR. The application is built on the SMARE code base (see section 6.1.1), and as such uses the same main menu and way of connecting to each other. The hints at the bottom of the screen is updated to reflect the placement of a skeleton rather than an anchor. The set up starts with the host detecting a plane and as the camera is looking towards a plane, a semi-transparent skeleton will appear and follow a small dot in the middle of the screen, as can be seen in figure 6.15. In addition to deciding where the skeleton should be placed, the host can do a pinch gesture to scale the skeleton up and down, and do a horizontal swipe to rotate it. The minimum size of the skeleton is half its original size and the maximum is twice the size. The semi-transparent skeleton, as well as the plane visualisation, will only be visible to the host.

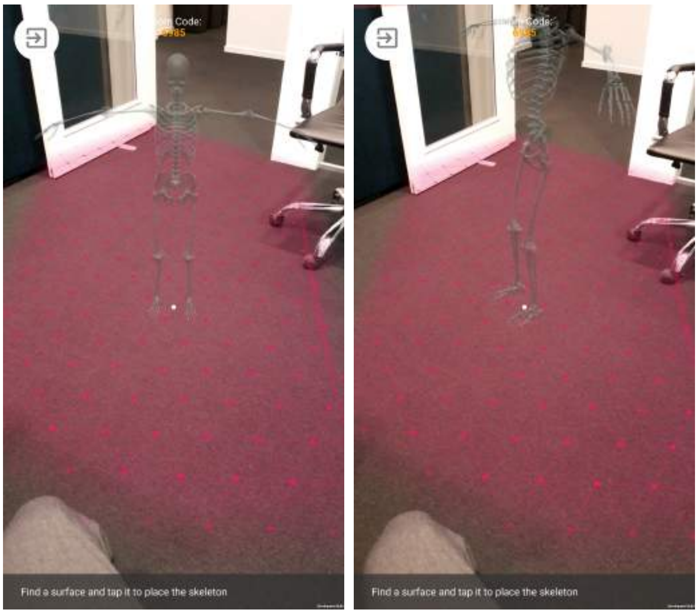

Figure 6.15: The skeleton can be scaled and rotated during placement.

Once the host is satisfied with the location, size and orientation of the skeleton, they can tap the screen to place it. At this point the semi-transparent skeleton will become fully opaque, the plane visualisation will disappear, a *Start Game* button will appear and an anchor will begin hosting at that location (see figure 6.16). Once it is successfully hosted, the onboarding messages will disappear for the host. The other users will then get a text and image telling them that they can begin resolving the anchor by looking from the host's angle at the area where the skeleton was placed. When it is successfully resolved, the skeleton will appear for the other users as well and the experience should be synchronised between the users.

When the host decides to start the game by clicking the button, the skeleton will be separated into several pieces that start falling down onto the ground, scattering all the pieces randomly across the floor. A piece can be either one big bone or several smaller bones grouped together. A semi-transparent skeleton will still be standing, indicating where the pieces should be placed (see figure 6.17). A piece can be picked up by the user by aiming at the item you wish to pick up and pressing somewhere on the screen. As the white dot in the middle of the screen passes over an item, that item will be be outlined in white to indicate that it can be interacted with. When an item is picked up, its position will stay relative to the device until the user looks at the semi-transparent skeleton, at which point the piece will automatically adjust its distance from the device to be the same as the skeleton. When the user releases their finger from the screen, they will drop the piece. If it is within range of the correct slot, it will place itself correctly, otherwise it falls to the ground. Bones will also cast a shadow on the ground beneath them to improve realism and depth perception.

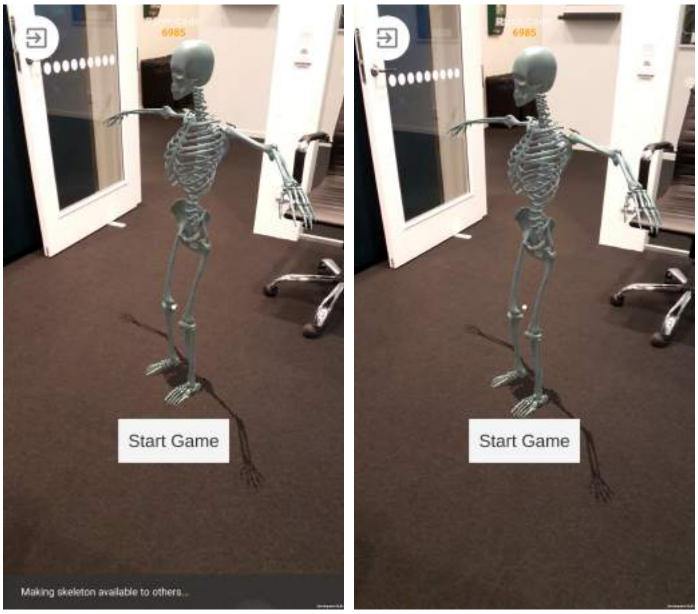

Figure 6.16: Once the skeleton is placed and hosted, the onboarding will disappear.

Figure 6.17: The user picks up a part of the spine and places it at the correct location.

If the other user picks up a piece, it will be highlighted in blue. In addition to this, the correct spot on the skeleton for that piece will also be highlighted in the same colour, as can be seen in figure 6.18. This way each user can give the other user hints when needed. While a piece is being held by a user, that piece is locked to that user and can not be picked up by other users. A piece that has been placed in the skeleton is set in place and can also not be picked up again. Once all pieces have been correctly placed and the skeleton is rebuilt, a label with the text "Puzzle Completed!" will appear in the middle of the screen.

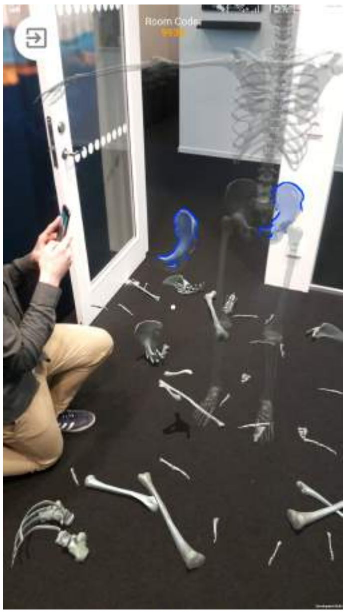

Figure 6.18: The other user holding a piece of the skeleton which is highlighted in blue.

6.2 Guidelines

The guidelines presented in this section are the final guidelines for this project. However, outside the scope of this project they should not be considered as finished or as an exhaustive list of guidelines for SMARE, but instead considered as areas of interest for further research within the area of SMARE. There are a total of 18 guidelines which have been divided into five different categories. The guidelines are not presented in any particular order and can be seen as an extension of already existing guidelines for AR, like the guidelines from Google [44] and Apple [75], and should as such be used in combination with general guidelines within AR as well as other relevant areas. In some cases though, the SMARE guidelines may contradict those other guidelines where the areas differ significantly. The presented guidelines are the result of prototyping and evaluating the three prototypes in this thesis. The documentation of the guidelines into their final form could also be considered an iteration as they have been further refined and expanded upon from the third iteration.

6.2.1 Environment

Taking the physical environment into consideration is an important part of SMARE and this category includes guidelines on *Experience Size*, *Object Scaling* and *Safety.*

6.2.1.1 Experience Size

Consider how many users the experience is intended for and how big area they need.

The amount of users and their interactions have a direct correlation to the amount of space which will be required for the experience. If the intention is to have many users in a smaller area, the experience should probably be designed around requiring less movement from the user as the area will be rather cramped. A smaller amount of users in a larger area allows for more movement without disrupting the experience, while a larger amount of users can fit in a smaller area if less movement is required. Taking these variables into consideration can help shape the experience for its intended use.

The Augmented Reality Design Guidelines [44] by Google, mention the importance of considering the size of the play area. AR-Showcase, AR-Racing and LumbAR have all shown that there are additional aspects worth considering, when designing for multiple users, which are directly correlated to the size of the area. It was first identified in AR-Showcase, where several users wanted the ability to freely walk around, resulting in a rather large area being used for the experience. In contrast to this, AR-Racing is an example where the users do not necessarily have to move around, which allows for a much smaller area to be used while having a relatively large amount of users partake.

6.2.1.2 Object Scaling

Giving users the ability to scale objects can help fit the experience in spaces of varying size and for a varying amount of users.

The places where users decide to engage in your experience may vary, some may want to place objects on the floor while others places them on smaller areas like tables. One thing that might affect their choice of place is how many users that are going to participate, as having many users will most likely require more space than for just a few. Also, some users may not have any large open spaces available. Letting users scale objects themselves can both help them fit the virtual content to the physical space that they have available and finding a content size big enough for all users to see.

In all the prototypes made during this project the users have been able to scale the content themselves. This has made it possible to use the applications in both smaller and larger areas. Feedback from the user tests has pointed in the same direction, that this is a good feature which makes the application work at a bigger variety of physical spaces and for a varying amount of users. Google is on the same track and writes in their AR guidelines that the areas may vary in size and that the experience should be responsive to how big the area is, but does not cover the implications of designing for a varying amount of users [44].

Figure 6.19: Trees scaled to different sizes in AR-Showcase, helping the user fit them into the physical space.

6.2.1.3 Safety

Be sure to keep all users safe during the entire experience.

With SMARE comes the implications of having several users engaging with an experience in a real environment. Consider where the experience could take place and how the users will engage in it. Even though AR allows the user to see parts of the real environment, as users get immersed in the experience, they might not have the same awareness of their surroundings but rather focus on the task at hand. If the intended area is in a potentially dangerous place, consider giving the users reminders that tells them to keep track of each other and their surroundings and/or give them the ability to define the experience area initially to avoid someone walking into a dangerous spot.

The evaluation of AR-Showcase brought up discussion on user safety and how it could be handled. This initially stems from guidelines by Apple [75] and Google [44], which touches upon considering the safety of the user. However, AR-Showcase and the other prototypes shows the added implications of having an AR experience designed for multiple users. An example of giving users warnings and reminders can be seen in Pokémon GO [4], where the user sometimes get notifications which reminds them to not trespass, enter dangerous areas or play while driving.

Figure 6.20: Example of a warning that is given in Pokémon GO [4].

6.2.2 Interaction Between Users

This category includes four guidelines concerning having two or more users interacting with each other in the same experience.

6.2.2.1 Verbal Communication

Consider how much verbal communication is required between users and how many users the experience is intended for.

If the experience is intended for a few number of users, it can be very beneficial to encourage verbal face-to-face communication [79]. However, having a large amount of users all talking over each other may cause a very chaotic and annoying experience. In this case it might be beneficial to have other ways of interacting that does not require as much talking.

For collaborative experiences similar to AR-Showcase and LumbAR, face-to-face communication can play an important role in enhancing the experience. While LumbAR is intended to be limited to two people at any time and uses verbal communication to great benefit, AR-Showcase can potentially be used by groups of 20 people or more. When used by a smaller amount of people, AR-Showcase works great for face-to-face communication but feedback implied that additional tools would help to improve communication when used by a larger amount of users simultaneously. An example of such a tool would be to give the user the ability to point at virtual objects via their device to reduce the need to verbally communicate what they are referring to. A feature similar to this can be found in Apex Legends [86], where the user can ping objects and locations, which will give a notification to team members and draw their attention to it.

6.2.2.2 Focus versus Face-to-face Interaction

Experiences which require more focus from the user tend to reduce the amount of face-to-face interaction between users.

Different applications require a varying amount of focus from the user, some needs the user to look at the screen all the time while others need the user to only interact with it once in a while. When doing an experience together, the users does not only interact with the devices, but also with each other. But the more focus the applications need, the less interactions it tend to be between the users. If the goal with your application is to get the users interact with each other it can be a good idea to design the app so the user can take their eyes of the screen sometimes without being punished by the system.

The different prototypes in this project have required a varying amount of focus. AR-Showcase and LumbAR did not need the user to look at the device all the time. This led to a lot of interactions between the users; they were talking, pointing out stuff with their hands, looking at each others screens and so on. On the other hand, AR-Racing was a prototype which really required the user to look at the screen all the time, otherwise you would with high probability crash the car while looking away. Therefore, during the user tests of AR-Racing, the users had a lot less interactions with each other, primarily a few comments to each other depending on what happened in the game. There was no pointing, looking at each others screens or such at all.

6.2.2.3 Collaboration and Social Interaction

If you want to encourage collaboration and social interaction between users, consider designing the experience in a way that they need to rely on each other.

Experiences can be made more enjoyable by having several people collaborating and socially interacting with each other. Simply giving users the ability to collaborate does not necessarily mean they will. To encourage collaboration and social interaction between users, consider giving users different tasks or asymmetric information in a way that they need to rely on each other in order to be effective or succeed.

LumbAR is a great example of how such an experience could be designed. In LumbAR, a user will get hints about where the other user should place their pieces, but no hints about their own. This gave the effect of users helping and communicating with each other in order to place the more difficult pieces. While it was not necessary for people to collaborate to complete the puzzle, it was enough encouragement to make the process more effective and it was utilised in order to complete the puzzle quicker.

Figure 6.21: In LumbAR, each user get hints for the other user to encourage collaboration.

6.2.2.4 Manipulating Same Object

Think about whether or not several users should be able to interact with the same object at the same time.

If several users try to modify an object in the same way at the same time, it may conflict with other users that are trying to modify the same object. This can make the users feel that they do not have control if they for instance are trying to move an object but it does not do what they expect, as someone else is also trying to move it. On the other hand it could also be used to create interesting interactions and behaviours between users which could be an important part of the experience.

During the development of LumbAR, there were problems of this kind where one user could steal a bone that an other user was holding. This was not a desired behaviour in LumbAR, since the user that lost the bone kind of felt that they had no control. In the final version, this was changed so a user had full control over a bone from the moment they grabbed it until they released it, and no one could steal it from them.

6.2.3 Sound

Some of the unique complications and possibilities of using sound in SMARE are covered by the four guidelines in this category.

6.2.3.1 Sounds for Many Users

Think of how many users your experience is designed for when deciding which types of sounds it should have.

Sounds can help immerse the user and enrich the experience [44], [75], but think of the intended number of concurrent users when designing the audio effects. If the experience is designed for a larger amount of users, consider using fewer, more distinct sounds as hearing the sounds from many other devices may cause confusion. An experience with fewer users can probably be designed with more freedom when it comes to sounds.

In AR-Racing, a cheering sound is played for a few seconds from your device every time you complete a lap. When multiple users complete laps at roughly the same time, sounds will be played from their devices simultaneously. This could cause confusion and be a pretty annoying experience if there are a lot of players making the sounds play continuously and overlap each other. The feedback received suggested that if there was many users in an experience, shorter and more distinct sounds could be used in order to make it less confusing and annoying.

6.2.3.2 Constant Audio

If constant audio is played asynchronously on several devices, it may be annoying or confusing and diminish the experience.

Try to make sure that when the same continuous sound is played from two or more devices, it is in sync. If the sound is not totally synchronised, it can be perceived as annoying by the user and instead give the opposite effect and diminish the overall experience.

During the user tests for AR-Racing, feedback was received about the sound that was coming from the cars. One of the things pointed out was that when the same sound was played on several devices while not being totally synced, the sounds became annoying due to the lack of harmony.

6.2.3.3 Sound Feedback

Consider using sounds as feedback for other users' actions.

Sounds can be used in a number of different ways to enhance the experience and improve the usability. If an experience is designed to have users within hearing distance of each other, this could be utilised to give useful feedback on different actions. Playing a sound on one device only can for instance give the users a hint that the user with that device has achieved or completed something. This has the benefit of not only notifying that a user has done something, but it also makes it easier to distinguish who it was and from where in the real environment that action was taken.

In AR-Racing, when a user completes a lap or the entire race, the sound of a cheering crowd is played from that user's device. This notifies the other users that someone completed a lap and they can hear from whose device the sound is coming from. Additionally, for the first user that completes the entire race, an additional fanfare sound is played from their device, indicating that they won.

6.2.3.4 Stereo Sounds

Having sounds that originate from virtual objects can help users understand where they are located.

Since there are more than one device in the room when using SMARE, stereo sound can be created. This can help users understand where an object with sound is located by playing its sound on all devices but with different volumes depending on how close they are to the virtual object. Since the different devices are not at the exact same position, the users can get hints about where an virtual object is just by listening to the sounds played by the devices.

In the second prototype AR-Racing, the cars had an engine sound attached to it that was played from the position of the car. Depending on where the car was on the track, the volume of the sounds differed between the devices and the users was therefore able to figure out roughly where it was by just listening.

6.2.4 Synchronisation

Having an experience that is synchronised between users is a core part of SMARE, and is also what the following four guidelines touch upon.

6.2.4.1 Accessibility versus Togetherness

Consider that individual manipulation of objects can increase the accessibility, but may diminish the users' feeling of togetherness.

The concept of SMARE includes being in the same physical space and seeing the same virtual objects. Usually people will be spread out during the experience and therefore not see everything from the exact same angle. Pointing out a specific spot on a specific object in a completely synchronised virtual environment requires the users to view the object from roughly the same position, to get a similar understanding. An alternative to this is to let the users locally manipulate their own instance of objects with actions like rotation, scaling and translation, without affecting the other users' objects. This can increase the accessibility for the users, making it easier for each user to see what they want to see. However, this also makes the experience less synchronised between the users' devices, which probably will diminish the feeling of togetherness, as users will no longer have a common understanding of where and how an object appears to be placed in the real environment.

In AR-Showcase, in order for everyone to see a specific country on the globe, everyone has to stand at the same place. Feedback from this prototype showed that users wanted to be able to locally rotate the objects to face them. In the Augmented Reality Design Guidelines [44] from Google they propose that the user should be able to rotate and move objects in order to increase the accessibility in cases where movement is restricted. However, these guidelines are mostly written for single user experiences. For SMARE though, letting people manipulate objects locally would still increase the accessibility but could also have a negative effect on the common experience through less synchronisation.

Figure 6.22: The user that sees this view can see Africa, while the user behind the globe can not without moving.

6.2.4.2 Information Visualisation

Consider visualising information such as text and instructions in a way that it is visible from any angle for all users who may need it.

Placing text in AR experiences can be done by either placing it directly on the display or by placing it inside the real world. For important information that is placed in the world, making sure it is always visible from any angle will make it more accessible. Also, consider that there are several users which may need access to the information and are looking at it from different angles. One way to achieve this is to always keep the text rotated towards each user, making the rotation unsynchronised between users. This could however also have the effect of decreasing the feeling of togetherness. In cases where placing a text in the world would make it inaccessible to some users, it might be beneficial to place it on the display instead.

Google provides a guideline on information visualisation and making text accessible to the user [44]. For SMARE however, all users must be taken into consideration. One approach to this can be seen in AR-Racing, where the race start countdown is displayed as a 3D-text placed at the start line and each user has their own instance of it which is rotated towards them. The feedback on this was that it was expected behaviour to have it rotated towards them and did not diminish the feeling of togetherness. Other information that was specific to one user, such as which lap they were currently on was instead displayed on the screen.

Figure 6.23: The countdown in AR-Racing is always facing the user, and the rotation is therefore not synced between the different devices.

6.2.4.3 Setup Instructions

Provide clear instructions to the users on how they should set up and join the experience.

Designing for SMARE means that there are some additional steps to the set up and these can easily go wrong. If the experience is not set up in the right way, problem with the synchronisation can occur. Make sure to give the users clear step by step instructions about how to set up and join the experience in the best way possible.

Instructions were not a thing that had high priority when implementing the prototypes, as there would always be somebody with knowledge to guide users if necessary. All the prototypes had a some helping text during the setup, and two of the prototypes also had images showing what to do for some steps. During the user tests of all prototypes, it happened several times that the users did not set up the application in the correct way, which resulted in a unsynchronised experience and users having the virtual content at different places in the real world. An example of an application that gives the user clear instructions is Just A Line [2], which for instance clearly tells the users to hold the devices side-by-side when connecting to the same session as this is optimal for ARCore [9].

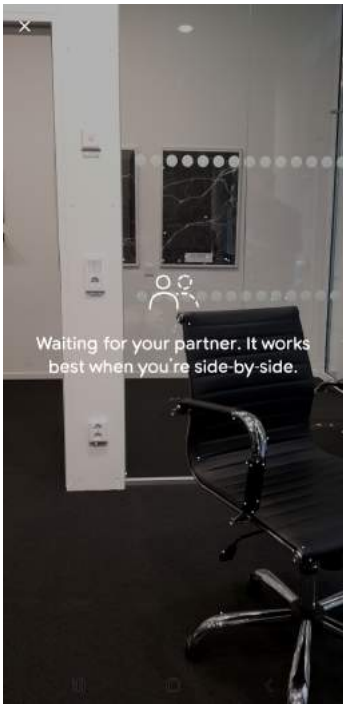

Figure 6.24: Instructions in Just A Line [2], showing what the user should do to set up the experience correctly.

6.2.4.4 Synchronisation Issues

Tell the user if something is going wrong and the experience is unsynchronised, and give them an easy way to restart or fix the problem.

There are always some things that can go wrong, such as a setup of the experience or losing tracking during the experience. In the case of SMARE, this could cause users to become unsynchronised and make virtual objects appear at different places. This is not always apparent to the users and may cause a negative experience. Try to identify those cases where something went wrong, inform the user and give them and easy way to restart or fix the problem.

The developed prototypes did not give any feedback to users when the experience was unsynchronised, which caused some issues for the users. In some cases they did not realise it was unsynchronised and wondered why others could not see the object they were pointing at. In certain cases it was obvious that something had went wrong and then they were usually quick to try to restart the setup and try again.

6.2.5 Miscellaneous

The following three guidelines did not really fit in any of the other categories but can be just as useful.

6.2.5.1 Active Users

Consider involving all users in the experience, as users who can not interact with the system for longer periods of time can lose interest.

Being part of a group of people engaging in an experience together can be quite enjoyable. Being an observer could also be enjoyable, but probably not for a very long time. It is usually more enjoyable to be actively involved in the experience. Therefore, it may be worth considering keeping all the users active during the experience. Having long periods of time without being able to interact with the system can make a user lose interest and get bored.

In AR-Racing and LumbAR, the users could interact with the system at any time. In AR-Showcase though, only the host could play around with the different models. The observers that were connected to the session could just look at the things that the host did, which they thought was cool in the beginning, but after a short period of time they did not think it was as fun anymore. Instead they wanted to also be able to play around just like the host did. This problem only occurred in AR-Showcase and not in the other two prototypes where all users were active all the time.

6.2.5.2 User Comfort

Think of ways to let the user take breaks without disrupting other users' experience to reduce the risk of users getting fatigued.

Carrying a mobile device for some time may get tiresome and what interactions the user is encouraged to do can have an impact on how long they can comfortably engage in the experience. Consider encouraging users to move their phone around or change the position in which they are holding it every now and then [44], [75], as doing the same movement over and over again can be exhausting for the user. Also, try to avoid forcing users into long play sessions by allowing a user to take breaks without disrupting the experience for others, or make it easy for users to all take a break simultaneously, pausing the experience.

The AR guidelines by Apple and Google mention the importance of user comfort [44], [75]. While testing LumbAR, the users felt a bit fatigued when having to move the phone repeatedly in the same way to pick up and place bones. Testing the prototypes also showed that it depends a lot on the individual how fast you get fatigued. Some experiences, such as AR-Showcase and LumbAR does not require everyone to be actively engaged the entire time, which allows for people to put down the device and take a break for a while without interrupting the experience for the others. AR-Racing on the other hand does require more constant focus during a race, but is instead designed around short races so that users can take a simultaneous break before they start the next one if they feel the need for it.

6.2.5.3 Depth Perception

Improve depth perception to reduce the risk of users perceiving objects to be at different locations.

Perceiving depth of virtual objects can sometimes be difficult and may cause users viewing the same object from different perspectives to perceive it differently. Use techniques such as occlusion and placing shadows beneath virtual objects to give better object presence and help the user perceive depth [44]. In cases where it is still difficult to perceive the depth of an object, consider designing the experience in a way that the users do not have to rely on the depth of an object to fully engage in the experience.

Figure 6.25: The solution in LumbAR for the problem with depth perception, where the piece the user holds snaps to the same depth as the skeleton when looking at it.

While this guideline originally stems from Google's AR guidelines [44], some additional implications of SMARE were identified in AR-Showcase. When talking with another user about a virtual object, there would sometimes be confusion as one user pointed at the object where it appeared to be for them while for another user looking at it from a different angle, it was either closer or further away. An example of how you can design the experience to not rely on depth perception can be seen in LumbAR, where objects held by the user will automatically adjust how far away they are to always be able to be properly placed at the correct distance, making depth a non factor.

7

Discussion

This chapter includes some thoughts and reflections on the process as a whole and the results in the form of prototypes and guidelines. Additionally there are some discussions about the validity of the guidelines, what parts of the project could potentially be interesting for related areas, and what are some ethical considerations related to this area. Finally there is a discussion on future work, what are some ways that the guidelines and prototypes could be developed further.

7.1 Process

The iterative process included creating three different prototypes, as we thought this would allow us to explore more aspects of SMARE rather than just having three iterations of the same prototype. This worked well for finding a big range of guidelines covering many different aspects and it was fun to create a few completely different experiences. Having only one prototype on the other hand would probably lead to a bit fewer but more refined guidelines more focused around what that specific prototype is about, in addition to an overall more polished prototype. Creating three different prototypes meant that a specific feature of one prototype could not be iterated upon with the feedback received from the evaluation of that user test, but was instead used when iterating on the guidelines as well using the lessons learned from the previous prototype when designing the new prototypes. This meant that while each prototype was isolated within one of the three iterations, they were still part of the overarching iterative process, focused primarily around the guidelines, and affected the outcome of all subsequent iterations.

The plan was to have three iterations of an equal size of four weeks each. This worked well initially and we followed that plan until we reached the start of the final iteration. At the start of the final iteration, we reviewed the time plan and took into account the dates for presenting the work, which was not available as the initial time plan was laid out. This showed us that the we would not have four weeks to finish the report, as was the initial idea, since the report had to be accepted one to two weeks prior to the presentation. This meant that in a worst case scenario we would finish the last prototype around the time the report should be handed in for review. With this knowledge and after some discussion with the supervisor, we decided to shorten the last iteration by a week which would hopefully give us two weeks to finish most of the report. Thankfully we had been continuously working on the report in parallel to working on the practical stuff which meant most of the process chapter was already written by the time we were done with the prototypes.

As mentioned, the plan was that each iteration would be four weeks long. Even though they would all be equal in size did not mean the prototypes could be equal in scope. Since the first iteration would include a lot of learning the tools and technologies used, the scope of the first prototype was kept small with a lot of optional features that could be added if there was enough time. For the subsequent iterations however, the scope increased as we had more knowledge which would hopefully allow us to implement more stuff in the same time. While it was true that the familiarity with the technology at this point in time did allow us to work faster, each iteration also presented some unique challenges which was not planned for. Regarding exactly what to create for each iteration was not something we had planned out completely beforehand. We had a rough idea about the first iteration, having one master user and the rest being observers seemed to be a good starting point to create something smaller in scope without a lot of interaction between users. We also had an idea that at least one of the prototypes could be a game, to explore that design space as well. Beyond that, what to create during each iteration was intentionally kept very open at the start of the project and time to develop such concepts was planned for the ideation session at the start of each iteration. The reason it was kept open was so that we could use the results of the previous iteration to help guide us in what direction we wanted to take the next prototype, since we could not know beforehand what feedback we would get from each prototype. For example, AR-Showcase did not have any sounds in it, yet during the user testing of that prototype there was a discussion about the use of sounds in SMARE, which led to us focusing on this more in AR-Racing.

The decision to extract the SMARE code base from the AR-Showcase was definitely the right decision. While this took some time away from the beginning of the second iteration, it would probably take even longer to redo the same implementation. More importantly, it saved us a lot of time for the third iteration, as the code base basically allowed us to get started right away and was probably one of the reasons we managed to complete a decent prototype in the shorter iteration time. The biggest time sink of implementing AR-Racing was by far the complications of mixing AR and advanced physics for driving the car. This issue took several days to solve, which if we would have known at the time would probably mean that we would try some other way to do it or even create something else, but the fact that we always seemed to be so close to solving it made it difficult to stop. We were basically at the point where we said that we were gonna try one last thing and if it does not work we have to do it in some other way, when we finally solved the issue and could continue where we left of a couple of days earlier. The actual implementation of LumbAR went really smooth, the biggest issue with this iteration was the shortened iteration time, which meant we had to cut some planned features. So while we did have more knowledge in subsequent iterations, which allowed us to work faster, each iteration presented its unique issues and challenges which in the end led to three prototypes of somewhat similar scope.

7.2 Results

This thesis has resulted in both prototypes and guidelines, which are both individually discussed in the sections below.

7.2.1 Prototypes

Three different prototypes; AR-Showcase, AR-Racing and LumbAR, have been developed during this thesis. Having in mind that every concept was ideated, prototyped and evaluated in three to four weeks, we are satisfied with the final results. The prototypes contained at least all the basic functionality that was vital for being able to user test it and gain some useful information to use in developing the guidelines, but in some cases even more than that was implemented. The primary goal of the prototypes was to make them good enough for testing and hopefully gain some insight into SMARE, and everything that was added beyond that was just a bonus.

All the prototypes use the SMARE code base, which is the reason that every prototype has identical main menus. We can agree on that the menu is not the most aesthetically pleasing we have seen, but since its visual appearance did not have any impact on what we wanted to get out of the user tests, we were satisfied with having it as it was.

One thing that the prototypes are lacking right now is good instructions of how to use the app. As it is now, the user has to play around and find out what is possible or not. For the purpose of testing though, we were fine with this since if the users did not understand something or had questions, they could just ask us. What caused most problems during the tests was that the prototypes did not have very detailed instructions on how to set up the Cloud Anchor [12] functionality. The users did not understand certain steps of this process, like holding the phones side-by-side, which affected the synchronisation in a negative way. These are probably the most important instructions that are missing. The actual tasks the the users could do after the set up was a lot more intuitive and most users managed to figure it out without instructions. During the test of AR-Racing, the users were told what they were going to do and what the controls were, but in AR-Showcase and LumbAR, they were not given that many hints. Still they managed to use the applications without any bigger problems. In LumbAR, the only thing that the users were told was that they should complete a puzzle, and they figured out the rest while using the app. In AR-Showcase the users also managed to quickly find out what the controls were and how they manipulated the objects in different ways. The fact that these not very polished prototypes with rather few instructions still are that intuitive to the user is something that we are very happy and satisfied with.

7.2.2 Guidelines

The thesis has not only been about developing prototypes, but even more importantly about delivering a set of guidelines for SMARE development. The final set consists of 18 guidelines, touching upon many different areas within SMARE. One thing to have in mind when reading these guidelines is that even though we call them final guidelines, they are only final for the sake of this project. We should not be considered experts in the area, and the guidelines have not been exhaustively tested. Therefore, the guidelines should be seen as interesting areas to explore further in future research, rather than finished guidelines. For developers and designers, these guidelines should be considered more as things that can be good to have in mind when designing SMAREs, rather than rules that tells you what you should do.

Most of the guidelines have been created as a result of feedback that we have received from the user tests or problems that have occurred during the development of the prototypes. When creating guidelines, there was often discussion about whether it was specific to SMARE or not. Some guidelines were pretty similar to what was already written in general AR guidelines from Apple and Google. Therefore we tried to find differences from their AR guidelines and additional implications of SMARE that we could apply to our own guidelines. For instance, Apple and Google have some guidelines on sound and that it can help immerse the user in the experience [44], [75], while in SMARE you will not only hear the sound from your own device, but also everyone else's, which means that the developer may have to think about these extra implications. The guidelines which we could see were different enough from general AR guidelines were added to the list, while the ones which were very similar and did not have anything new for SMARE, were discarded.

7.3 Validity

There are several factors which have affected the outcome of the guidelines and may have an impact on the validity of them. The guidelines were primarily shaped around the developed prototypes and developing other types of prototypes would surely have an impact of what guidelines they would result in. While the aim was to keep guidelines as general as possible to potentially be applied to any SMARE, they have only been tested on three prototypes at most and there may definitely be exceptions found to these guidelines depending on what the experience is.

The guidelines are to a big part based on the evaluation of the prototypes and as such how the user tests and evaluation of them was conducted may have had an impact on the final guidelines. The methods used, the specific questions asked and also the people participating in the tests most likely affected the outcome and doing it in some other way would probably yield another result. Also, while you try to be as objective as possible when evaluating the user test and summarising observations and feedback, people might interpret things differently and its therefore hard to make sure that it is completely objective.

In our user tests, there were often the same kind of people that participated, mostly students from Chalmers and employees at Inceptive and Uptive. This is something that could have affected the outcome of the user tests to be different compared to if we had done tests on a broader variety of users. Also, some users did participate in all three user tests which also could have affected the result in a certain way since they would have more experience in AR, even though they did not have any previous experience in using that specific prototype. We made sure though that we for each prototype had at least a few users that had not tried any previous prototype to not use the exact same set of users in all tests. Even though our study is not focusing on a user group with a specific amount of experience, it could not hurt to have some people of varying experience. The number of users in each test was also roughly the same, from two to three people. We did see some differences between the tests of two people and three people, mostly regarding the size of the space that was needed. To get better data from the tests, tests with for instance 10 users at the same time could have been useful, and even if we did that, tests with an even higher number of users could have been made in order to improve the results. There were some limiting factors such as the amount of compatible devices that prevented us from doing tests of this size within the scope of this project.

While the results of using the guidelines have been tested through the prototypes, the guidelines have only been applied to the design by us. Since the guidelines have not been used by any other designers for designing SMAREs, they might be tailored around our process and not yield the same results for other designers. Since we developed the guidelines, we have an understanding of how they should be used. Other designers may interpret and apply the guidelines in another way or not understand how to use them.

7.4 Generalisability

Although the prototypes was developed using ARCore Cloud Anchors [12] and specifically for mobile phones and tablets, the goal was to have guidelines that could be applied to SMARE, no matter which specific technology or platform that is used. AR-glasses is one example of an up and coming new platform for AR where these guidelines may be of interest. With the pace that AR is developing currently, it would be unfortunate to have the guidelines be outdated as soon as a new platform becomes mainstream. How important or relevant a specific guideline is may vary depending on platform to some degree but hopefully they can be somewhat useful in at least indicating interesting areas worth considering.

As the SMARE code base have been used during the project to create three different prototypes, it could also be used as a starting point for creating any sort of SMARE using these specific technologies. A lot of the stuff might need to be changed to fit the specific requirements of the experience but it could work as a starting point, much like the Cloud Anchor sample application it is based upon, which we started with.

7.5 Ethical Considerations

ARCore Cloud Anchors [12] was used to create the prototypes. To make the Cloud Anchors technology work, Google need to collect data that is gathered through the users' cameras. According to Google themselves, the data collected is removed seven days after hosting an anchor [87]. This is nothing that developers using Cloud Anchors can do anything about. The data collected is used to resolve anchor requests by other devices so they can be paired. However, Google is not very clear about if the data can be used for something else. They do write though that "It is not possible to determine a user's geographic location or to reconstruct any images or the user's physical environment" [87], so developers simply have to trust Google that they do not use the data inappropriately.

Nowadays there are sometimes discussions on whether we are using our phones too much or not, and how that can affect our everyday life. It is not uncommon to see people on the streets that are looking at their phones ignoring everything that happens around them. This may not be the biggest problem if the user is walking around in a big empty area, but if done in more crowded places or close to traffic it can be dangerous. It is possible to argue that this problem is not going to be as big with AR applications as with normal mobile applications, since the user actually sees the real world through the phone. At the same time, the virtual AR objects on the screen are covering parts of the picture so important details in the surrounding can be missed anyway. Also, focusing on an application on a phone, even if it is an AR application, will make the user less aware of what is happening around them, outside of what is visible to them on the screen.

Another subject that is worth discussing is the fact that the AR objects are only visible on the screen. This means that everyone who does not see the screen, especially those who are not up-to-date with the technologies, may think that the user is using the camera to take pictures or record videos. Even though this is not intended by the user, it can still make people uncomfortable if they think they are being recorded.

While the accessibility of AR is increasing, it still requires relatively new devices to work. This fact combined with the multi-user aspect of the domain, could have a social impact on groups of people that wish to use the technology when not everyone in the group is able to participate because they have an older device. Depending on the context in which such an application would be used, it could cause a gap between groups of people which is worth considering. Another similar problem is that everyone cannot afford to buy a new phone that is compatible with AR, which can make AR a technology only available to the wealthier people.

7.6 Future Work

This section covers the discussion of what could be done if additional time and resources were spent at further developing the prototypes and the guidelines of this study. The future work of the prototype will mostly cover aspects which could be improved upon or added to make it a better product rather than a better prototype for further testing the guidelines. What further work could be done to the guidelines is instead covered under subsection 7.6.2.

7.6.1 Prototypes

As has been previously mentioned, time spent on developing good instructions and onboarding was kept to a minimum as we could communicate these verbally during the user tests. If these prototypes were to be developed further into actual products that would reach users outside of the project, they would definitely need improvements on that part. They would all benefit more having a more specific and clearer onboarding to setup the experience, but also some instructions or hints on how to use the application and what they can do. When it came to figuring out how to use the application, users often managed to figure out pretty quickly what interactions they could do but in the case that a user could not figure it out, it would probably be beneficial to provide some kind of help that is optional and not too intrusive to be annoying for people that do not need it. AR-Racing was probably the one users had most issue with figuring out how to use properly, while AR-Showcase and LumbAR seemed to be more intuitive.

7.6.1.1 AR-Showcase

There were some things like environmental light estimation on virtual objects and ground shadows which we had not yet figured out during the first iteration and experimented more with in the later prototypes. Going back to AR-Showcase, these are things that you can really notice are missing and would be beneficial additions. There are also some more major aspects which would need further work. Since the prototype was developed simply for finding guidelines, there were only four random models which we had added for testing purposes and there was no back end functionality for adding your own 3D-models to display. As such, the prototype would provide little value for an actual user that would like to present something in AR. If this was to be published, it would definitely need the ability for users to upload their own 3D-models.

7.6.1.2 AR-Racing

For AR-Racing there were a few different things which we were considering adding if time would allow for it. The first one, which we also covered in the process, is powerups. Powerups did not make the cut as we could not manage to implement them in a way that felt good with the time we had, but something we still believe could be a cool addition if we had the time to do it right. It would be a great tool to increase the interaction between players which we think would lead to a more enjoyable game. In addition to powerups, we would also like to add a few different maps which the host can decide between when placing it. This would allow for a greater variation in gameplay and would also allow the users to better utilise the space they have available in the real world by choosing a track that fits good. Different maps could be designed for a different number of users and have additional elements like speed boosts and hazards placed throughout the track. Potentially, we could even add the ability for users to build their own tracks by connecting building blocks such as curves and straights.

7.6.1.3 LumbAR

One of the initial plans for LumbAR was to have different difficulties, but this was cut due to the shorter iteration cycle. As the human skeleton has over 200 bones, we had to group bones together to make it a decent experience. Ideally, we would like to have an easy, medium and hard mode. Where hard would more or less mean placing each individual bone, medium would be roughly about the level which LumbAR is at now and easy to mean bigger groups of bones like placing an entire hand at once for example.

LumbAR could potentially be seen as a learning experience where you learn about the location of the bones in the human body. We would also like to in some way integrate the name of the bones in the puzzle somehow. Either by just displaying the name of the bone your holding or by making them a vital part of the puzzle, perhaps another game mode. This could potentially also teach the user about the name of the bones in addition to where they are located. The model used for the bones also has other parts of the human body available like the nervous system, muscles and similar. These could potentially be used to add additional puzzles that are related to the human body.

7.6.2 Guidelines

Further work on the guidelines would include making a more exhaustive list of guidelines and improving the validity of them by further working on the shortcomings discussed in section 7.3. This would mean testing the guidelines on a wider range of experiences for several platforms and technologies within the scope of SMARE, having other designers use the guidelines when creating SMAREs, and doing more user tests on a wider variety of users.

8

Conclusion

This thesis identifies and defines the area of Synchronised Multi-user Augmented Reality Experience, SMARE. This domain refers to AR experiences where several users can see and interact with the same virtual objects at the same location in real time, while the objects appear in a similar state to all users. The purpose of the thesis has been to explore this newly identified area by answering the following research question:

What aspects should be considered when designing synchronised multi-user augmented reality experiences?

The question is answered through 18 guidelines that have been produced during this project. To come up with the guidelines, three concepts have been ideated, prototyped and evaluated in order to gather information about the area of SMARE. Guidelines were created iteratively from the knowledge, experience and feedback received from these prototypes.

The results of the thesis is partly the three prototypes; AR-Showcase, AR-Racing and LumbAR, and partly the guidelines answering the research question. AR-Showcase is a tool for presenting 3D-models in AR by placing the virtual objects in your physical environment. AR-Racing is a racing game for AR in which the user places a virtual race track on a physical surface and play together with up to three friends. In LumbAR, the goal is for the user to, alone or together with a friend, piece together a skeleton in AR. The guidelines are touching upon many different areas within SMARE and a total 18 guidelines was created, divided into five different categories.

One category is *Environment*. In this category there are guidelines that are about the surrounding areas of the user, and what to consider regarding that. Environment includes the following three guidelines:

- *Consider how many users the experience is intended for and how big area they need.*
- *Giving users the ability to scale objects can help fit the experience in spaces of varying size and for a varying amount of users.*

- *Be sure to keep all users safe during the entire experience.*

The next category is *Interaction Between Users*, which contains guidelines about users and how they interact with each other, both face-to-face and through the system. The four guidelines in this category are:

- *Consider how much verbal communication is required between users and how many users the experience is intended for.*
- *Experiences which require more focus from the user tend to reduce the amount of face-to-face interaction between users.*
- *If you want to encourage collaboration and social interaction between users, consider designing the experience in a way that they need to rely on each other.*
- *Think about whether or not several users should be able to interact with the same object at the same time.*

The third category is *Sound*, which includes guidelines about the new challenges when using sounds in SMAREs. The following guidelines are placed in this category:

- *Think of how many users your experience is designed for when deciding which types of sounds it should have.*
- *If constant audio is played asynchronously on several devices, it may be annoying or confusing and diminish the experience.*
- *Consider using sounds as feedback for other users' actions.*
- *Having sounds that originate from virtual objects can help users understand where they are located.*

Synchronisation is the next category. It contains guidelines that are about the synchronisation of devices and how it can affect the experience. The guidelines in this category are:

- *Consider that individual manipulation of objects can increase the accessibility, but may diminish the users' feeling of togetherness.*
- *Consider visualising information such as text and instructions in a way that it is visible from any angle for all users who may need it.*
- *Provide clear instructions to the users on how they should set up and join the experience.*
- *Tell the user if something is going wrong and the experience is unsynchronised, and give them an easy way to restart or fix the problem.*

The last category is *Miscellaneous*, which contains the guidelines which did not fit in any other category. The three guidelines in this category are:

- *Consider involving all users in the experience, as users who can not interact with the system for longer periods of time can lose interest.*

- *Think of ways to let the user take breaks without disrupting other users' experience to reduce the risk of users getting fatigued.*
- *Improve depth perception to reduce the risk of users perceiving objects to be at different locations.*

The guidelines should be considered as interesting areas to consider when designing SMAREs, rather than strict rules. The guidelines would need more research and testing before they can be considered as an exhaustive list of correct and valid guidelines for designing SMAREs. The prototypes should also not be considered as finished products, even though they were good enough for this project, but would need to be further developed in order to be released to the public as a finished product. Mixtive, the company which most of the thesis work was carried out at, may possibly develop the prototypes further and integrate them into their projects.

Bibliography

[1] WCS100, "Louis Rosenberg Augmented Reality Rig," 2014. [Online]. Available: https://commons.wikimedia.org/wiki/File:Louis_Rosenberg_Augmented_Reality_Rig.png

[2] Google LLC, "Just a Line - Draw anywhere, with AR." [Online]. Available: https://justaline.withgoogle.com/ (Accessed 2019-01-23).

[3] Warpin Media, "ARrrrrgh." [Online]. Available: https://www.warpinmedia.com/cases/arrrrrgh/ (Accessed 2019-02-06).

[4] Niantic Inc. and The Pokémon Company, "Pokémon Go." [Online]. Available: https://www.pokemongo.com/en-gb/ (Accessed 2019-02-11).

[5] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Trans. Information Systems*, vol. E77-D, no. 12, pp. 1321–1329, 1994. [Online]. Available: https://www.researchgate.net/publication/231514051_A_Taxonomy_of_Mixed_Reality_Visual_Displays

[6] Vector Unit, "Beach Buggy Racing." [Online]. Available: https://www.vectorunit.com/beach-buggy-racing (Accessed 2019-04-05).

[7] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001. [Online]. Available: http://ieeexplore.ieee.org/document/963459/

[8] Apple Inc., "ARKit." [Online]. Available: https://developer.apple.com/arkit/ (Accessed 2019-01-03).

[9] Google LLC, "ARCore." [Online]. Available: https://developers.google.com/ar/ (Accessed 2019-01-03).

[10] Google LLC, "Supported Devices," 2018. [Online]. Available: https://developers.google.com/ar/discover/supported-devices (Accessed 2019-01-03).

[11] Apple Inc, "Device Compatibility," 2017. [Online]. Available: https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMatrix/DeviceCompatibilityMatrix.html (Accessed 2019-01-03).

[12] Google LLC, "Share AR Experiences with Cloud Anchors," 2018. [Online]. Available: https://developers.google.com/ar/develop/java/cloud-anchors/overview-android (Accessed 2019-01-03).

[13] Inceptive AB, "Inceptive." [Online]. Available: http://www.inceptive.se/ (Accessed 2019-02-06).

[14] Mixtive AB, "Mixtive." [Online]. Available: https://www.mixtive.com/ (Accessed 2019-02-06).

[15] Mixtive AB, "AR Call." [Online]. Available: http://arcall.ly/ (Accessed 2019-02-06).

[16] Mixtive AB, "SpotBuild." [Online]. Available: http://www.spotbuild.com/ (Accessed 2019-02-06).

[17] Mixtive AB, "AR School." [Online]. Available: http://www.arschoolyard.com/ (Accessed 2019-02-06).

[18] Mojang Synergies AB, "Minecraft." [Online]. Available: https://minecraft.net/ (Accessed 2019-02-06).

[19] ArtoolkitX, "artoolkitX." [Online]. Available: http://www.artoolkitx.org/ (Accessed 2019-01-29).

[20] Vuforia, "Vuforia - Engine." [Online]. Available: https://www.vuforia.com/engine.html (Accessed 2019-01-29).

[21] Interaction Design Foundation, "Augmented Reality – The Past, The Present and The Future," 2018. [Online]. Available: https://www.interaction-design.org/literature/article/augmented-reality-the-past-the-present-and-the-future (Accessed 2019-01-23).

[22] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 341–377, jan 2011. [Online]. Available: http://link.springer.com/10.1007/s11042-010-0660-6

[23] Microsoft Corporation, "Microsoft HoloLens." [Online]. Available: https://www.microsoft.com/en-us/hololens (Accessed 2019-01-23).

[24] Magic Leap Inc, "Magic Leap One." [Online]. Available: https://www.magicleap.com/magic-leap-one (Accessed 2019-01-23).

[25] Glass, "Glass." [Online]. Available: https://x.company/glass/ (Accessed 2019-01-23).

[26] Vuzix, "Vuzix Blade® | Augment Reality (AR) Smart Glasses for the Consumer." [Online]. Available: https://www.vuzix.com/products/blade-smart-glasses (Accessed 2019-01-23).

[27] O. Bimber, R. Raskar, and R. Raskar, *Spatial Augmented Reality.* A K

Peters/CRC Press, aug 2005. [Online]. Available: https://www.taylorfrancis.com/books/9781439864944

[28] Unity Technologies, “Products - Unity.” [Online]. Available: https://unity3d.com/unity (Accessed 2019-01-24).

[29] Unity Technologies, “Company - Unity.” [Online]. Available: https://unity3d.com/company (Accessed 2019-01-24).

[30] Unity Technologies, “Unity - Manual: Integrated development environment (IDE) support,” 2018. [Online]. Available: https://docs.unity3d.com/Manual/ScriptingToolsIDEs.html (Accessed 2019-01-24).

[31] Unity Technologies, “Unity - Learn.” [Online]. Available: https://unity3d.com/learn (Accessed 2019-01-24).

[32] Unity Technologies, “Unity Multiplayer.” [Online]. Available: https://unity3d.com/unity/features/multiplayer (Accessed 2019-03-15).

[33] Exit Games, “PUN.” [Online]. Available: https://www.photonengine.com/en-US/PUN (Accessed 2019-03-15).

[34] Exit Games, “Multiplayer Game Development Made Easy | Photon Engine.” [Online]. Available: https://www.photonengine.com/en-US/Photon (Accessed 2019-03-15).

[35] Google LLC, “ARCore Cloud Anchors.” [Online]. Available: https://codelabs.developers.google.com/codelabs/arcore-cloud-anchors/index.html?index=..%2F..io2018#0 (Accessed 2019-01-24).

[36] H. W. J. Rittel and M. M. Webber, “Dilemmas in a general theory of planning,” *Policy Sciences*, vol. 4, no. 2, pp. 155–169, jun 1973. [Online]. Available: http://link.springer.com/10.1007/BF01405730

[37] W. Gaver, “What should we expect from research through design?” in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI ’12*. New York, New York, USA: ACM Press, 2012, p. 937. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2207676.2208538

[38] Google LLC, “Google Drive.” [Online]. Available: https://www.google.com/drive/ (Accessed 2019-01-24).

[39] Atlassian Corporation Plc, “Trello.” [Online]. Available: https://trello.com/ (Accessed 2019-01-24).

[40] Electronic Arts Inc, “FIFA 19.” [Online]. Available: https://www.easports.com/uk/fifa (Accessed 2019-01-24).

[41] Epic Games Inc, “Fortnite.” [Online]. Available: https://www.epicgames.com/fortnite/en-US/home (Accessed 2019-01-24).

[42] Blizzard Entertainment Inc., “World of Warcraft.” [Online]. Available: https://worldofwarcraft.com/en-us/ (Accessed 2019-01-24).

[43] K. J. Bergström and Karl, “Playing for togetherness : designing for interaction rituals through gaming,” Ph.D. dissertation, Göteborg University, apr 2012. [Online]. Available: https://gupea.ub.gu.se/handle/2077/29072?locale=sv

[44] Google LLC, “Augmented Reality Design Guidelines.” [Online]. Available: https://designguidelines.withgoogle.com/ar-design/ (Accessed 2019-02-04).

[45] M. Billinghurst, I. Poupyrev, H. Kato, and R. May, “Mixing realities in Shared Space: an augmented reality interface for collaborative computing,” in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 3. IEEE, 2000, pp. 1641–1644. [Online]. Available: http://ieeexplore.ieee.org/document/871085/

[46] L. Yangguang, L. Yue, and W. Xiaodong, “Multiplayer Collaborative Training System Based on Mobile AR Innovative Interaction Technology,” in *2014 International Conference on Virtual Reality and Visualization*. IEEE, aug 2014, pp. 81–85. [Online]. Available: http://ieeexplore.ieee.org/document/7281047/

[47] L. Frankel and M. Racine, “The Complex Field of Research: for Design, through Design, and about Design,” in *DRS Montreal—Design and Complexity*, 2010, pp. 518–529. [Online]. Available: http://www.drs2010.umontreal.ca/data/PDF/043.pdf

[48] B. Schneider, “Design as Practice, Science and Research,” in *Design Research Now*. Basel: Birkhäuser Basel, 2007, pp. 207–218. [Online]. Available: https://www.springerlink.com/index/10.1007/978-3-7643-8472-2_12

[49] M. Mohseni, B. Liebold, and D. Pietschmann, *Game Research Methods [Lankoski & Björk]*. lulu.com, 2015.

[50] W. C. Booth, G. G. Colomb, J. M. Williams, J. Bizup, and W. T. FitzGerald, *The craft of research*, 4th ed. University of Chicago Press, 2016.

[51] B. Martin and B. Hanington, *Universal Methods of Design: 100 Ways to research complex problems, develop innovative ideas, and design effective solutions*. Beverly, MA: Rockport Publishers, 2012.

[52] B. Moggridge, *Designing Interactions [Hardcover]*. MIT Press, 2007. [Online]. Available: https://mitpress.mit.edu/books/designing-interactions

[53] R. Dam and T. Siang, “5 Stages in the Design Thinking Process.” [Online]. Available: https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process (Accessed 2019-01-25).

[54] J. Preece, Y. Rogers, and H. Sharp, *Interaction design : beyond human-computer interaction*. Chichester, West Sussex: John Wiley & Sons Ltd, 2015.

[55] J. Nielsen, *Usability Engineering.* AP Professional, 1993.

[56] Mind Tools Content Team, "Brainstorming." [Online]. Available: https://www.mindtools.com/brainstm.html (Accessed 2019-02-05).

[57] M. Diehl and W. Stroebe, "Productivity loss in brainstorming groups: Toward the solution of a riddle." *Journal of Personality and Social Psychology*, vol. 53, no. 3, pp. 497–509, 1987. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/0022-3514.53.3.497

[58] R. Dam and T. Siang, "Introduction to the Essential Ideation Techniques which are the Heart of Design Thinking," 2019. [Online]. Available: https://www.interaction-design.org/literature/article/introduction-to-the-essential-ideation-techniques-which-are-the-heart-of-design-thinking (Accessed 2019-02-05).

[59] Assistant Secretary for Public Affairs, "Prototyping," feb 2014. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/prototyping.html (Accessed 2019-01-25).

[60] T. Fullerton, C. Swain, and S. Hoffman, *Game design workshop : a playcentric approach to creating innovative games.* Elsevier Morgan Kaufmann, 2008.

[61] D. Mortensen, "Best Practices for Qualitative User Research," 2019. [Online]. Available: https://www.interaction-design.org/literature/article/best-practices-for-qualitative-user-research (Accessed 2019-02-05).

[62] "Manifesto for Agile Software Development," 2001. [Online]. Available: https://agilemanifesto.org/ (Accessed 2019-01-24).

[63] J. Sutherland and K. Schwaber, "What is Scrum?" [Online]. Available: https://www.scrum.org/resources/what-is-scrum (Accessed 2019-01-03).

[64] K. Beck and C. Andres, *Extreme Programming Explained: embrace change*, 2nd ed. Addison-Wesley, 2005.

[65] M. M. B. Poppendieck and T. D. Poppendieck, *Lean Software Development: an agile toolkit.* Addison-Wesley, 2003.

[66] B. Meyer, *Agile, The Good, the Hype and the Ugly.* Springer International Publishing, 2014.

[67] K. Schwaber, *Agile Project Management with Scrum.* O'Reilly Media, Inc, 2009.

[68] L. Rising and N. Janoff, "The Scrum software development process for small teams," *IEEE Software*, vol. 17, no. 4, pp. 26–32, 2000. [Online]. Available: http://ieeexplore.ieee.org/document/854065/

[69] Google LLC, "Google Scholar." [Online]. Available: https://scholar.google.se/ (Accessed 2019-03-06).

[70] IEEE, "IEEE Xplore Digital Library." [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp (Accessed 2019-03-06).

[71] DiGRA, "DiGRA Digital Library." [Online]. Available: http://www.digra.org/digital-library/ (Accessed 2019-03-06).

[72] ACM, "ACM Digital Library." [Online]. Available: https://dl.acm.org/ (Accessed 2019-03-06).

[73] Chalmers Library, "Chalmers Library." [Online]. Available: http://www.lib.chalmers.se/ (Accessed 2019-03-06).

[74] Y.-N. Chang, R. K. C. Koh, and H. Been-Lirn Duh, "Handheld AR games — A triarchic conceptual design framework," in *2011 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities*. IEEE, oct 2011, pp. 29–36. [Online]. Available: http://ieeexplore.ieee.org/document/6093653/

[75] Apple Inc., "Human Interface Guidelines." [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/ios/system-capabilities/augmented-reality/ (Accessed 2019-02-22).

[76] J. Johnson, "Fingers Lite - Free Finger Touch Gestures for Unity," 2019. [Online]. Available: https://assetstore.unity.com/packages/tools/input-management/fingers-lite-free-finger-touch-gestures-for-unity-64276 (Accessed 2019-03-22).

[77] J. Johansson, "ARCoreUtils," 2018. [Online]. Available: https://github.com/jonas-johansson/ARCoreUtils (Accessed 2019-03-22).

[78] J. Nielsen, "How Many Test Users in a Usability Study?" 2012. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/ (Accessed 2019-03-15).

[79] M. Billinghurst and H. Kato, "Collaborative augmented reality," *Communications of the ACM*, vol. 45, no. 7, jul 2002. [Online]. Available: http://portal.acm.org/citation.cfm?doid=514236.514265

[80] Team17 Group PLC, "Overcooked." [Online]. Available: https://www.team17.com/games/overcooked/ (Accessed 2019-03-22).

[81] Wikipedia, "Mario Kart," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Mario_Kart (Accessed 2019-03-22).

[82] Unity Technologies, "Unity Asset Store." [Online]. Available: https://assetstore.unity.com/ (Accessed 2019-04-05).

[83] Rorije, "Dust & Rust Racetrack Pack," 2016. [Online]. Available: https://assetstore.unity.com/packages/3d/vehicles/dust-rust-racetrack-pack-9385 (Accessed 2019-04-05).

[84] VDGAMES, "Cartoon Sport Cars - Unity Asset Store," 2018. [On-

line]. Available: https://assetstore.unity.com/packages/3d/vehicles/land/cartoon-sport-cars-60618

[85] Wikipedia, “Rock Band,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Rock_Band (Accessed 2019-04-29).

[86] Respawn Entertainment, “Apex Legends.” [Online]. Available: https://www.ea.com/en-gb/games/apex-legends (Accessed 2019-05-08).

[87] Google LLC, “Share AR Experiences with Cloud Anchors.” [Online]. Available: https://developers.google.com/ar/develop/unity/cloud-anchors/overview-unity (Accessed 2019-02-08).

A

User Tests Iteration One

A.1 First Test

The first test was conducted on 3 participants simultaneously. It was carried out in a lunch room where the two of us and the three testers could fit without any problems. The test was done mostly around the table, which was pretty big, with sufficient space for moving freely around it. All testers remained standing throughout the entire test and moved around continuously.

Observations:

- Användarna verkade direkt förstå konceptet med rumskod och delade med sig av den muntligt
- Hosten blev disconnectad av okänd anledning
- Hade velat ha en indikator på att man väntar på hosten (typ timglas)
- Alla står/sitter still att börja med medans rummen skapas upp och kompassen placeras ut.
- Bytte bord på grund av problem med trackingen på enfärgad/reflektivt bord
- Hosten följer instruktionerna i onboardingen
- En användare som inte förstod animationen i onboardingen (om man faktiskt skulle göra rörelsen)
- Förstod inte direkt vad man ska göra direkt efter kompassen är placerad
- Efter några sekunder hittar de plusset men försöker placera ut objekt genom att klicka. Efter några försök så inser hon att man kan dra ut objekten.
- Testade direkt att börja manipulera objektet genom olika bekanta gester, lyckades dra runt, skala och rotera utan problem.
- Försökte individuellt att rotera objektet som observer, men insåg sen att de inte kan manipulera objektet.
- Insåg efter ett tag att man kan dölja plane visualization, och att den kommer

upp som hjälp för att placera objekt.

- Alla förflyttar sig runt bordet för att se från olika vinklar
- En användare tappar emellanåt tracking när man kollar bort från kompassen, kommer tillbaka när man tittar mot kompassen.
- Lyckades ta bort objekt utan större problem (såg ingen dra för att ta bort, endast klicka)
- En disconnectade men kunde komma in snabbt igen genom att fråga efter koden

Byte av roller, annan master user.

- Förslag: Sprida koden via t.ex. apple tv eller skriva upp på en tavla
- Råkade sätta ut kompassen utan vetskap, antog att den placeras ut automatiskt
- Flera testade att manipulera även kompassen
- Ibland va det lite problem att försöka flytta specifika objekt (träffade inte objektet)
- Försökte lämna rummet och joina igen för att rätta till trackingen
- Stod inte på samma plats när man letar efter kompassen så de blev osynkade
- Förslag: Förtydliga att man kan dra in objekt

Scenario: Visa sydafrika

- Alla ställer sig på typ samma ställe till en början för att se från samma vinkel
- Förslag: Visa info om plats när man trycker på jordklotet
- Förslag: Lägga till gravitation (och physics?)
- Kraschade på en telefon av oklar anledning
- Förslag: Visa antal användare i rummet för att kunna få en snabb överblick över om alla är med i rummet, eventuellt visa namn på deltagare med
- Förslag: Spela in “lektionen” så att man kan gå tillbaka och se vad man diskuterade
- Select-circles göms under bordet pga. occlusion

Intro questions

Have you used AR applications before?

Ja på två, lite på 1

If yes - Have you used many? Can you give examples of which apps you

have used?

Person one: Möbler och mätningsverktyg. T.ex. ikeas

Person two: Snapchat, 3-4 appar

Person three: Massa olika. Mät och kommunikationsverktyg. ARCall och andra appar som inceptive/uptive gjort.

Social Interaction

Did you feel like there were aspects of the application that either succeeded or failed to support co-located collaborative work? What are those aspects and in which way did they succeed/fail?

Alla ser samma sak, får ett jordklot i 3d som man kan gå runt och kolla. Hade velat ha någon interaktion för eleverna. Kanske bara för sig själv typ rotera. Alla måste som det är nu stå på samma plats för att se samma sak. Hade eventuellt velat kunna rotera sin egna jordglob, annars måste alla kunna stå på samma plats för att se samma del av jordgloben. Sätta ut pins på jordgloben, för att kunna peka ut platser/länder även om alla har egna rotationer. Om man individuellt kan ändra så borde det finnas en knapp för att resetta så man återställer till master clients läge. Kanske kunna slå på ett lås så att eleverna kan göra saker själva. Ett mode där alla ser samma och ett där alla kan interagera. Alla kanske skulle kunna samarbeta och bygga upp en kultur i klassrummet tillsammans.

Was something missing in either the case of the host or the observer to help with the social interaction between users?

Kunna peka ut grejer. Ändra sitt egna state. Trångt om alla ska stå på samma ställe för att se samma sak. Kunna sätta egna origon, typ vid sina egna bänkgrupper. Hur hanterar man kommunikation mellan väldigt stora grupper av människor? Hur ställer man frågor utan att prata i mun på varann när man är många. Kunna göra sina egna saker och sen skicka en request till läraren för att visa resten av klassen. Typ som en presentation.

Was there any features that diminished the presence and face-to-face interaction between the users?

Tycker att dem pratade ungefär som om dem hade tittat på fysiska objekt. Blev att dem försökte ha lite ögonkontakt. Kände inte att dem blev avskärmade från resten.

User Safety

Did you feel like there were interactions that could be dangerous depending on the environment and number of people?

Typ utomhus kan det bli farligt, typ som i pokemon go att folk går ut mitt i vägen. Stora saker kan man behöva se från långt avstånd. Skulle man t.ex. vara på en fästning så kanske man ramlar ner någonstans. Bra med varningsmeddelande. T.ex. att välja en bra omgivning. Om man behöver röra sig på större ytor, kan det bli jobbigt om t.ex. barn behöver röra sig i skolan, vad händer om barn går så långt att dem försvinner. Varningar om man rör sig för långt från origo. Kanske avgränsa ytan innan det startar för att det inte ska spåra ur under upplevelsens gång.

Förslag: att kunna välja att köra på sin egen plats.

Could you think of what such interactions/actions could be? How would you design around those?

-

Play Area

Did the application work well in this environment? Did you feel like the area was too small or too big?

Svårt att säga då det var mer lek än en specifik uppgift. Bra med skalningsfunktion, så kan man anpassa objekt efter utrymmet.

Do you think the application could work well in different sized areas, what aspects allow for this or limit this? Any suggestions?

Att det går att zooma gör att det fungerar på flera olika områden. Mer interaktioner med objekt, typ att man kan öppna motorhuven på en bil eller dela jorden. Begränsa hur man kan skala objekt för att ha interaktiva modeller.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Ändra egna modeller lokalt, som sagt tidigare. Duplicera på ett enklare sätt. T.ex. om man har skalat objekt och vill ha ett i samma storlek. Ändra färger på objekt.

Did you find a way to overcome this restraint? How?

Gå runt bordet när man inte kunde rotera själv. Kan även fråga "läraren" att rotera modellen. Förslag: läraren kan trycka på en knapp som får alla att titta på samma sak. "Sync view". Läraren kanske på sikt ska kunna välja mellan "alla ser samma" eller "alla kan göra sitt eget lokalt".

Were you able to access all necessary information at all times?

Önskas: Antal användare, se usernames på dem som var inne. Då kan man märka om någon har problem eller om någon gör något annat på ipaden. Kanske få poäng om man vart inne länge. Möjlighet att kunna betygsätta lektionen så läraren får feedback. Hade vart intressant att kunna köra detta på olika platser. Att man på två olika platser kan se samma objekt och visa för varandra.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Nej, men kan tänka sig att det blir jobbigt om fler (säg en klass på 20) ska se samma sak och måste knö på samma plats. Ju fler användare desto mer behövligt blir det att kunna modifiera sina egna objekt. Eller typ dela upp det på fler bord. Kan bli jobbigt att hålla telefonen under längre sessioner.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

-

Feedback

Did you feel that the system gave you enough feedback on your actions? When did it? When didn't it?

-

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Ja, objekten synkades bra och med en gång så länge de drogs ut i scenen. Kanske kan lägga till att alla kan se vilket objekt som läraren har markerat så alla vet vad man pratar om. Markera med antingen cirkeln eller att objekt flyter i luften. Om flera kan interagera så kanske man kan lägga till att användarnamn syns på selectionen så man vet vem som interagera med objektet.

Lägga till ljud för att få en bra experience?

General

Did you feel that the application gave you enough information of how to use it? Were there any functions that you didn't notice?

Vara tydligare när saker går fel. Kanske en lite större text just när det går fel. Annars var det inga större frågetecken. Haptisk feedback kanske kan hjälpa i vissa fall. Kan förstärka upplevelsen.

What did you feel about the interactions of the app? Were they easy to figure out or not? Were there any confusing or unclear interactions?

-

Did you understand all parts of the interface? Was it clear which functions the different parts represented?

-

Placing a model and deleting a model were some features available as part of the 2D interface while scaling, rotating moving models were done by directly manipulating virtual objects. Do you think it would be beneficial/disadvantageous to move any of these features between the 2D and 3D interface? Any ideas of how that could be done?

Tyckte det var bra som det var. Hade bra kontroll. Ska man ha många objekt så blir listan lite liten. Kanske isåfall kan få upp något objekt i 3D som man kan interagera med för att få fram modellerna. Så kan läraren där välja t.ex. vilket ämne osv. Listan störde inte så mycket i telefonen, bra att man kan dölja den.

How do you think this app would work if there was 5, 10 or 20 people in a session? Any ideas on how to support a larger amount of users?

-

Övrigt:

Ljud, kanske att läraren styr och triggar ljud på allas enheter? Kan funka så länge det inte är någon fördröjning mellan enheterna. Tre olika varianter att välja på, Cloud anchors, fritt eller same view as teacher.

A.2 Second Test

The second test was conducted on 2 people simultaneously. The test was carried out in a smaller office where the two of us and the two testers could practically just fit, which resulted in a very limited area to move around on. The area utilised in the test was a small table, about 50x50cm. The participants remained seated on their chars, on opposite sides of the table, through most of the experience.

Observations:

- Trycker på create room men inget händer (ingen internetanslutning)
- Andra användaren skapar ett rum medans den första ansluter till internet
- Värden börjar röra telefonen enligt animationen, lyckas snabbt och inser att det är ett plan som motsvarar bordet.
- Deltagaren frågar efter rumskod och hosten delar med sig av den verbalt.
- Antar att man kan välja ett plan när "Tap the surface to start" va framme på skärmen. Sätter sedan ut kompassen.
- Deltagaren ser också animationen för att skanna in kompassen och gör vad den visar men lyckas inte hitta kompassen. (Sitter på sin plats mitt emot värden, alltså kollar från fel vinkel)
- Efter uppmaning från oss så ställde sig deltagaren bredvid värden och lyckades därmed hitta kompassen.
- Värden trycker på plusset och ser att han kan dra ut modeller till planen och förstod att krysset stängde ner menyn.
- Tryckte på ett plan och planvisualiseringen försvann, förstod inte riktigt hur han gjorde det dock.
- Deltagaren rör sig för att se objekten från olika håll. Tryckte på ett objekt

och det flög iväg (Problem med tracking?)

- Värden drar runt objekt som finns i världen.
- Deltagaren byter till att hålla mobilen i liggande läge.
- Kompassen har flyttat på sig för värden men han förstår att den är "så att man kan relatera".
- Värden inser också att han kan dra objekt till papperskorgen.
- Undrar vad som händer om han lämnar rummet, fick upp confirmation dialogen som bekräftade vad han tänkte.
- Deltagaren tycker att pilen på ikonen för att lämna rummet pekar åt fel håll.

Byte av roller.

- Värden tycker att prickarna är intuitiva och förstår att de motsvarar ytan på bordet, även om den råkat tracka planet större än vad bordet egentligen är. Förstår att olika plan indikeras med olika färger.
- Värden sätter ut en kompass och försöker sedan att lista ut hur man döljer planvisualiseringen men misslyckas.
- Deltagaren påpekar att han lyckades dölja planet men vet inte hur han gjorde det.
- Värden börjar tycka att prickar är irriterande. Inser att han kan skala och rotera objekt genom olika gester.
- Försöker placera objekt ett objekt i luften men inser att det inte går. Hade velat placera objekt på olika höjder.
- Försöker skanna sin hand för att få en yta högre upp.
- Inser att han har markerat ett objekt och att han kan ta bort det genom att klicka på papperskorgen, men vet inte vilket objekt som är markerat och kommer tas bort.
- Trycker på papperskorgen och ser att planet försvann. Testar även att dra ett objekt till papperskorgen som han precis dragit ut för att inte placera det. Använder papperskorgen som en lösning för att dölja planytan.
- Deltagaren förflyttar sig runt i rummet för att se de utplacerade objekten.

Uppgift: Visa sydamerika för deltagaren

- Värden börjar med att skala upp en jordglob som är placerad på bordet mellan dem.
- Letar efter sydamerika, hittar det och snurrar sedan globen ett halvt varv för att visa det för deltagaren. Värden tog dock fel på sydamerika och sydafrika.

- Deltagaren flyttar sig till värden för att fixa trackingen då kompassen har förflyttat sig. Löser trackingen och går sedan tillbaka.
- Jordgloben är roterad så deltagaren kan se rätt kontinent från sin egna position.
- Värden döljer återigen planet genom papperskorgen. Insåg snart att ytan visas när objekt är markerade.
- Båda går ut ur appen.

Intro questions

Have you used AR applications before?

Ja och nej

If yes - Have you used many? Can you give examples of which apps you have used?

Snapchat AR filter för båda.

Social Interaction

Did you feel like there were aspects of the application that either succeeded or failed to support co-located collaborative work? What are those aspects and in which way did they succeed/fail?

Det va bara en person som fick göra grejer så det va inte så mycket samarbete. En som fick göra grejer och en som fick titta på.

Was something missing in either the case of the host or the observer to help with the social interaction between users?

Hade eventuellt velat kunna sätta en markör på jordgloben som en observer ifall målet är samarbete. Ifall det är en presentation eller föreläsning så vill man inte att den andra ska kunna göra sånt men då är det inte heller collaborative. För att använda utrymmet bättre så kanske man inte vill vara begränsad till planen i vart man placerar grejer. Kanske kunna flytta objekt vertikalt. Kan t.ex. få fram pilar för att kunna förflytta i en viss riktning, är dock också bra som det är nu att slippa ha olika modes för skalning, förflyttning etc. Ska användaren ha mer kontroll så krävs det även mer interface etc.

Was there any features that diminished the presence and face-to-face interaction between the users?

Fokuserade på mest på skärmen och inte så mycket på den andra användaren. Kände inte att det fanns något behov i scenariot att titta på den andra användaren. Kände av närheten av den andra användaren. "Blir mindre men inte skadligt mindre"

User Safety

Did you feel like there were interactions that could be dangerous depend-

ing on the environment and number of people?

Ifall alla 20 pers ska springa runt och kolla på sydafrika så kommer det bli kaos. Kan ha två lägen. Läraren antingen styr jorden för att visa en specifik plats så alla så de ser samma eller så kan alla snurra sin egna version av jorden bara för att titta allmänt.

Could you think of what such interactions/actions could be? How would you design around those?

Designa upplevelsen för att alla inte ska behöva förflytta sig runt väldigt mycket, desto fler personer desto mer stillastående. Är det större grupper så dela upp det så inte alla tittar på samma grej.

Play Area

Did the application work well in this environment? Did you feel like the area was too small or too big?

Miljön fungerade okej för att de bara var två stycken. Hade man varit fler så hade det behövts större utrymme. Desto fler objekt man satte ut desto trängre kändes det. Va också lite jobbigt att man va tvungen att gå runt bordet för att se kompassen.

Do you think the application could work well in different sized areas, what aspects allow for this or limit this? Any suggestions?

Om utrymmet är alldeles för stort så behöver man skala upp objekt väldigt stort för att alla ska kunna se allting. Systemet supportar inte väldigt små grejer väldigt bra. Ifall man lägger ut många små saker så måste man backa bak med mobilen för att få en överblick.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Ville kunna ta bort grid i början, som observer ville man kunna göra grejer.

Did you find a way to overcome this restraint? How?

Kommunicerade verbalt och försökte peka i verkligheten men då behöver man synka med skärmen. Djup är lite svårt att uppskatta.

Were you able to access all necessary information at all times?

Det va otydligt i onboarding hur man skulle hitta kompassen. Försökte bara röra mobilen runt men behövde gå runt bordet. Va tvungen att unselecta en model först innan man kunde lämna rummet, vilket inte va vad man förväntade sig.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Gjorde lite knasiga rörelser för att manipulera objekten.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

Att hålla en viss position en längre tid kan vara jobbigt. Kan tänka sig att man kollar ett tag, sen tar ner mobilen några minuter och sen tittar igen. Telefonbatteriet hade klagat om inte annat. Kan va bra med korta pauser.

Feedback

Did you feel that the system gave you enough feedback on your actions? When did it? When didn't it?

Va lite otydligt vilket objekt man hade markerat. Cirkeln under va inte alltid i vy.

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Hade ingen aning om den andra användaren placerade ut objekt förens han sa att det va objekt där. Att allting uppdaterades i realtid gjorde det tydligt att man förflyttade, skalade objekt osv. Visste inte hur många som har joinat rummet och hur många som har synkat.

General

Did you feel that the application gave you enough information of how to use it? Were there any functions that you didn't notice?

Synca från samma håll. Fanns inte info om vad man kunde göra. Var börjar mina möjligheter och vart slutar dom? Det kändes intuitivt hur man rotera, skala osv. Visste dock inte om man kunde förflytta vertikalt. Trial and error. Sålänge det är intuitivt så kan trial and error vara helt okej. Jobbigt dock om det kan finnas ett annat sätt att göra det på som man inte upptäcker. Gillar inte långa tutorials.

What did you feel about the interactions of the app? Were they easy to figure out or not? Were there any confusing or unclear interactions?

-

Did you understand all parts of the interface? Was it clear which functions the different parts represented?

-

Placing a model and deleting a model were some features available as part of the 2D interface while scaling, rotating moving models were done by directly manipulating virtual objects. Do you think it would be beneficial/disadvantageous to move any of these features between the 2D and 3D interface? Any ideas of how that could be done?

Ta bort knappen skulle man kunna flytta till 3d världen. När man markerar ett objekt så kan det komma upp att det går att ta bort den. Ser inget specifikt som skulle vara bättre eller sämre.

How do you think this app would work if there was 5, 10 or 20 people in a session? Any ideas on how to support a larger amount of users?

Beror på scenariot. . . ?

Övrigt

Man har nästan glömt interfacet för att man har pratat så länge.

B

User Tests Iteration Two

B.1 First Test

Observations:

- Startade med en gång, har ingen koll på vems bil är vem.
- Startade utan ljud
- Upptäckte att man kan backa genom att hålla inne bromsen.
- Har problem att styra när man backar.
- Båda kommenterar sina drifts och kollisioner med varandra.
- En person förflyttar sig för att titta från en annan vinkel. Tappar tracking.
- Måste hålla devicen konstigt för att inte täcka kameran.
- Gör ett nytt rum för att en person disconnectade.
- Råkar trycka på skärmen och placera tracken av misstag.
- Försöker byta/ändra bana men går inte.
- Råkar hålla finger i vägen.
- De testar att sitta på golvet istället
- Lyckas inte resolva banan på rätt ställe.
- Startar nytt spel genom att klicka rematch.
- Kör in i väggar. Tycker det är svårt att ta sig loss.
- Väggarna känns som lim, borde sänka friktionen.
- En har gått i mål men den andra fokuserar mest på att bara krocka med den andra.
- En av dem råkar hålla fingret över kameran hela tiden
- Kör ett nytt race

- En försöker fuska genom att backa men lyckas inte.
- Kör rakt in i motståndaren som flyger av banan.
- Verkar ha olika strategier; en försöker mest förstöra för den andra som försöker komma först i mål.
- Håller fingrar för halva kameran utan att märka det i några sekunder.
- Blir svettiga om händerna.

Båda höll vid flera tillfällen fingrarna framför kameran utan att flytta dem.

Intro questions

Have you used AR applications before?

Har använt snapchat filters och den första prototypen

If yes - Have you used many? Can you give examples of which apps you have used?

Se ovan.

Play Area

Did the application work well in this environment? Did you feel like the area was too small or too big?

Nee, man vill ha ungefär sån här plats. Behövde inte så mycket mindre eller större. Man kunde skala banan för att anpassa. Blir den för stor känns det som man måste gå upp och förflytta sig för att se allt. Känns bekvämare att ha en mindre som man kan se hela.

Do you think the application could work well in different sized areas? Why/why not? Any suggestions?

Kan tänka sig att man känna sätta upp som en stor racing-bana med större publik runt om. Kommer dock va svårare att styra den för att man inte kommer ha lika bra översikt i svängarna längre bort. Underliggande problemet är att man inte kan se grejer bra som är långt borta, man skulle kunna få en liten annan vy som ett bilspel men det förstör lite syftet med appen. Kanske kan ha osynliga väggar för att de inte ska täcka vyn.

Social Interaction

Did you feel like there were any elements of the game that encouraged social interaction between users? Was something missing?

Att man kunde krascha med varandra och interagera med varandra i spelet främjade. Om man kör in i en vägg så man hamnar på olika delar av banan så kan man inte interagera med varandra. En lösning på det kan vara powerups som gör att man kan interagera med varandra vart man än är. Hade nog främjat den sociala interaktionen.

Audio

Did you feel like the sounds enhanced or diminished the experience? What specific sounds and in what way?

Förbättrade upplevelsen, även det jobbiga ljudet visar att något händer. Hade dock stängt av det efter tag. Ljud på countdown va bra. Tyckte inte motorljudet ändrade någonting.

En reagerade på den andras ljud för att han råkade hålla för högtalarna. Märkte att det bara kom ljud på den personens device.

How do you think the use of audio would affect the experience for a larger/smaller amount of users?

Om alla ska ha surrande ljud så hade det blivit jobbigt med många personer. Också ljudet när man går i mål hade man hört hela tiden. Hade nog i så fall fått se till att använda enstaka ljud då och då snarare än att de kommer hela tiden.

Any ideas on how audio can be used to enhance the experience in multi-user AR applications?

Ljudeffekter när man krockar med varandra. Om man skjuter en missil så hör man ljud från den personen som skjuter och kanske från den som blir träffad.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Va obekvämt att hålla mobilen på ett konstigt sätt för att inte täcka kameran etc samtidigt som man skulle styra bilen. Började bli trött i armarna efter ett tag som man började hålla telefonen längre ner men då insåg man att man inte såg hela banan längre. Gäller att ha en bra position från början.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

Är nog samma problem som ovan men man hade nog också lärt sig hur man ska hålla den. Placering av kontroller på skärmen. Om man har synkat in banan på ett sätt så måste man hålla den positionen sen.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Att backa och svänga va jobbigt. Man va tvungen och tänka 2-3 ggr när man backade. Hade velat ha powerups. Hade kanske velat hantera hastigheten själv, typ ha sliders för hur mkt man svänger. Tänkte först att bromsarna backade i olika riktningar.

Did you find a way to overcome this restraint? How?

Gjorde tvärt om vad man tänkte när man skulle backa, då gick det oftast.

Were you able to access all necessary information at all times?

Saknade feedback på när man tryckte på knappar på skärmen eller inte. Hade varit kul att se vilken hastighet. Utöver att se att bilen svänger sig så hade velat veta att devicen registrerar det man gör.

Feedback

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Hade ingen aning vilket lopp den andra personen va på. Kan se vart han är på banan.

General

What did you feel about the interactions of the app? Were there any confusing or unclear interactions?

Att man måste kolla från hostens synvinkel för att synka det. Va osäker på vilka begränsningar som fanns och vad som orsakar problem.

The lap counter was displayed on the screen of the device while the countdown for the race was displayed in 3D. Do you think it would be beneficial/disadvantageous to move any of these features between the 2D and 3D interface? Any ideas of how that could be done?

Laps borde vara på skärmen. Hade det varit i världen så känns det som information som alla har. Det som finns i världen är sånt alla ser medans sånt på skärmen är det bara jag ser. Om alla ska se all laps så kan det kanske placeras i världen.

Tänkte inte så mycket på att texten va riktad mot en, kändes naturligt. Hade varit konstigt annars.

B.2 Second Test

Observations:

- Host skapar ett rum och delar med sig av koden.
- Host hamnade i menyn igen, så han skapar ett nytt rum.
- Host skalar och placerar en bana på ett bord.
- Båda är fokuserade och lever sig in i spelet, typ säger neej när det går dåligt
- En spelare prejar den andre av banan
- Glömmer bort hur man backar

- Ena spelaren går i mål och vinner
- Båda verkar ha problem vid en sväng där dem ofta har problem
- båda står helt still under racen
- nytt race
- inte så mycket trashtalking, mer att man ropar saker när det går dåligt
- En spelare tycker det är “inverterat” att backa (syftar antagligen på svängningen)
- Spelaren som inte vann race 1 vinner race 2

Intro questions

Have you used AR applications before?

Användare 1 - nej

Användare 2 - lite olika men typ ikea eller apples slingshot

If yes - Have you used many? Can you give examples of which apps you have used?

-

Play Area

Did the application work well in this environment? Did you feel like the area was too small or too big?

Tyckte banan var lite för stor från början, men skalade ner den så att den skulle passa. Det var lite jobbigt att håller fingret undan kameran. Man glömde ibland att följa med med kameran när man körde ur bild. Det kändes lite pluttigt och svårt och se, mobilskärmen är ju inte så stor.

Do you think the application could work well in different sized areas? Why/why not? Any suggestions?

Skulle kunna göra så att banan autoscalar efter hur stor ytan är. Banan kan vara större än det man ser men då blir det jobbigt att hänga med med handrörelser, vilket kan vara jobbigt när man håller telefonen i ett jobbigt grepp. Ha en typ karta eller annan vy där man kan se var bilen är om man har ett inzoomat läge.

Social Interaction

Did you feel like there were any elements of the game that encouraged social interaction between users? Was something missing?

Krockarna. Kan lägga in namn på bilarna, funkade nu när man var två men är man fler så blir det svårt att veta vem som är vem. En topplista där man ser placeringarna hela tiden. Hade nog velat ha den som UI, 3d är mer spelet medan highscore känns mer 2d. Kanske ha en leaderboard med varvtider ute i världen. Kanske kan utnyttja ar och göra en stor prispall som visas bredvid banan så blir det tydligt vem som leder. Typ som ett hologram.

Audio

Did you feel like the sounds enhanced or diminished the experience? What specific sounds and in what way?

Ljuden som fanns är väl den typ av ljud som man vill ha. Måste kanske inte finnas men det kan förhöja upplevelsen. Lite coolt/weird/ovant att bilen låter mer när den är nära kameran.

How do you think the use of audio would affect the experience for a larger/smaller amount of users?

Det hade blivit stökigt. Blir brötigt med mycket ljud på samma gång. Kanske kan synca ljuden så att det blir mer harmoni.

Any ideas on how audio can be used to enhance the experience in multi-user AR applications?

Tror det är viktigt att tänka på harmonin när man är många så det inte blir för brötigt.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Använde inte backen så ofta, det var lättare att respawna. Det får inte bli för mycket kontroller. Man fick flytta mobilen lite för att kunna se bilen hela tiden, speciellt när den var långt bort. Vid större banor kanske det blir att man får springa runt för att kunna hänga med. Kanske inte ha höga väggar för att man ska kunna se.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

Hålla telefonen i rätt läge kan bli jobbigt i längden. Man vill kunna sätta sig ner efter ett tag men då ser man inte så bra. Kör man på golvet kan man välja om man ska stå eller sitta. Det är jobbigt med AR allmänt, att stå och hålla en device såpass länge. Tror det blir bättre när glasögon kommer.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Kändes som att backen inte fungerade ibland. Backen kändes lite onödig, räcker med respawn. Då blir det mindre kontroller och enklare att köra.

Did you find a way to overcome this restraint? How?

Respawna. Backa runt i en hel cirkel. Lite svårt när man inte fick feedback på svängar.

Were you able to access all necessary information at all times?

Varvtider. Se namn och positioner på andra om man är många. Kunna se hur många varv dem andra har kvar. En tutorial i början som visar hur man gör hade varit bra.

Feedback

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Inte mer än att man såg den andres bil, men man fokuserade mer på sitt eget race. Kunde vart mer när den andra passerade mål osv.

General

What did you feel about the interactions of the app? Were there any confusing or unclear interactions?

Nej, men bilen kändes väldigt hal i kurvor och så. Kändes inte så naturligt när man körde in väggen heller. Lite oklart när man svängde eller inte, hade velat ha mer feedback på det.

The lap counter was displayed on the screen of the device while the countdown for the race was displayed in 3D. Do you think it would be beneficial/disadvantageous to move any of these features between the 2D and 3D interface? Any ideas of how that could be done?

Kanske skulle kunna ha något mitt på skärmen där det finns utrymme. Kan komma upp stora siffror som räknar ner på något roligt sätt. Sånt som finns hela tiden typ highscore skulle nog passa i 2d, medan typ specialeffekter som typ nedräkningen skulle passa bra i 3d. Skulle kunna tänka sig att ha typ egen information i världen, man märker ändå inte att den inte är syncat.

Bilstyrningen var lite konstig. Få feedback från bilen när man svänger typ att hjulen svänger.

C

User Tests Iteration Three

C.1 First Test

Observations:

- Skapar ett rum, delar med av kod.
- Host placerar skeleton på marken.
- Kan klicka start, frågar om han andra är redo.
- “Okej, tror vi ska bygga upp honom”
- Testar att dra benen till spöket.
- Försöker rotera benet, vill kunna rotera det efter man plockat upp det.
- Håller i en bit och försöker lista ut vad man kan göra.
- “Aha, jag kan se vart din bit ska sitta”
- Märker att den roterade sig när man sätter den.
- -Vi säger att det inte går att rotera manuellt i nuläget.
- “Måste bara försöka på samma ställe flera gånger för att den ska fastna”
- Hjälper varandra med vart grejerna ska sitta.
- En användare “fuskar” en gång genom att titta på den andra skärmen när de står bredvid varandra.
- “Menar du mitt höger eller hans höger?”
- Måste sätta dit varje finger
- En användare förflyttar sig en del för att plocka upp ben och placera dom rätt, går ofta närmare och byter vinkel.
- Den andra användaren står mest still under hela upplevelsen.
- Vill kunna kolla närmare på benen men det får man inte.

- “Detta är en sten. Nej det är en knäskål.”
- Det är så jäkla många grejer över allt.
- Vet inte om jag orkar med alla dessa fingrar och tår.
- Det här fingret är väldigt långt borta så det blir väldigt litet, men om jag går nära så får jag det lite större.
- Måste vara väldigt pricksäker på små fingrar, hade kunnat vara generösare med hitboxen.
- Är vi klara? Borde ligga någon lite tå någonstans antagligen.
- Klarar spelet.
- Gick framför varandra en del.

Intro questions

Have you used AR applications before?

Har använt våra tidigare AR-Applikationer

If yes - Have you used many? Can you give examples of which apps you have used?

Se ovan.

Interaction

How well do you think the interactions used to pick up and place pieces of the skeleton worked? Why? (Depth. . .)

Att plocka upp grejer fungerade okej på de stora grejerna. Det första man försökte göra va att ta o dra i skeletten. Så fort man drog pricken över så highlightade den ett ben då förstod man att det va en cursor. En person klickade hela tiden i mitten av skärmen för att plocka upp medans den andra bara klickade vart som på skärmen.

Va jobbigt att man inte fick någon indikation på vart den skulle sitta, vilket man kan förstå, men om det va fel va man tvungen att böja sig ner o ta upp det igen. Vill kunna rotera för att se om den kan passa in på den platsen och sen få någon form av feedback.

Försökte skala upp ett objekt men det gick inte. Tyckte det va bra att det “snappade” till objektet.

Tyckte det va jobbigt att den va långt bort när man plocka upp den men bra att den la sig på rätt avstånd sen.

Feedback

Did you feel that the system gave you enough feedback on other people’s actions? When did it? When didn’t it?

Om man ville ha koll så kunde man ha koll. Man fick lista ut själv att den visade vart den andres bit skulle sitta. Systemet förklarade inte. Vill inte ha det in your face men kanske att man har olika färger på highlights så det är tydligt.

Social Interaction

Did the fact that each user had information that could help the other user enhance or diminish the experience?

Det va bättre, det blev mer av en gruppaktivitet. Kan även säga att den ska vara på vänster sida när den ska vara på höger så det blir lite kul.

Could you think of any other features to encourage social interaction between users?

Kanske om man måste ta varannan bit eller något. Då när jag inte lägger en bit så kan jag inte göra något annat än att hjälpa dig. Blir mer regler vilket kan va jobbigt men om man vill öka interaktionen.

Tycker ändå att det va en rimlig mängd interaktion mellan användarna. Vissa delar vet man och vissa vet man inte. Blev ofta att man försökte placera något och ifall det inte gick så frågade man den andra.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Att behöva kolla upp och ner hela tiden. Mobilen blev väldigt varm.

Kanske kan lösa genom att biten man håller på med nu är “low gravity”. Att man har ett ben man fokuserar på nu men om man tar upp ett nytt så faller det gamla ner.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

Va även lite jobbigt att hitta grejerna. Va också jobbigt att de va så små delar som man va tvungen och sätta ut. Hade det varit ett pussel med 6000 bitar så.

Känns inte som det blir svårare att ha många delar, det blir bara jobbigare. För att göra det mindre jobbigt så borde man ha färre lika delar. För att göra det svårare borde man inte ha fler delar utan istället göra något anna koncept. T.ex. ha fler nivåer: efter skelett så kan man till exempel köra inälvor och så.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Ja. -Rotation på benet -Kunna flytta benet i djupled -Vill kunna “inspecta” benet utan att gå genom hela processen att ta upp det och dra det närmare sig. Inspecta som att typ det kommer upp ett fönster ovanför resten av skärmen, där det står info om benet samt kunna rotera och kolla på endast det benet.

Om man kommer till organen och så sen så vill man kanske ha genomskärningar för doktorand studenter.

C.2 Second Test

Observations:

- Väntar på host att placera skelett
- Tänker att man ska plocka upp delar och placera dem
- Såg att när man sikta på dem så kan man plocka upp dem
- Försöker röra/dra delen med fingret men det går inte
- Man vill rotera den på något sätt men det går inte.
- Wow! det va snyggt när den satte sig på plats
- Funderar på varför man inte ser blått på sina egna delar
- Förstår sedan att man ser varandras ledtrådar
- Funderar på vart man ska placera delarna men använder fortfarande inte ledtrådar
- Kollar efter om man får någon feedback när man placerar
- Appen kraschar.
- Varför visar den två händer? (i onboarding)
- Blir osynkade positioner på skeletten men det verkar de inte inse och kör på
- Det är "motståndarens" ledtrådar.... eller ska man samarbeta?
- Börjar ge varandra ledtrådar.
- Tar turer i att ge ledtrådar.
- Börjar sedan istället ta en del för att visa den andra vart den ska sitta, sen släpper man den och plockar upp själv när man vet var den ska sitta.
- Tycker att delen placeras snyggt när man sätter dit den
- Hjälper varandra att justera höjd och så på delarna för att sätta dit dem.
- Står mest stilla på sina platser, justerar sig lite ibland för att få en bättre överblick/se från lite annan vinkel.
- Pratar mycket för att placera delarna.

- Letar runt efter någon sista bit. Förflyttar sig runt för att titta runt efter delar.
- Puzzle completed!

Intro questions

Have you used AR applications before?

Användare 1 - nej

Användare 2 - lite olika men typ ikea eller apples slingshot

Båda här även testat AR-Racing

If yes - Have you used many? Can you give examples of which apps you have used?

Se ovan.

Interaction

How well do you think the interactions used to pick up and place pieces of the skeleton worked? Why? (Depth...)

Tycker det fungerade ganska bra faktiskt, va ändå ganska intuitivt. Man såg att det highlightades och tryckte på skärmen för testa så märkte man att det gick. Den andra försökte dra i benbiten men lärde sig ganska snabbt. Försökte sätta i lårbenet men det va helt fel vinkel från skelettet så tänkte att man försökte rotera det för att få det att passa in. Det fick okej att uppfatta djupet men förstod inte varför delen hoppade riktigt ibland. Kändes som man tappade kontroll. Va jobbigt att tappa benen och kändes som att de kom längre bort än de borde (svårt att uppskatta djup)

Feedback

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Ja, man såg när de andra benbitarna är i luften och det blåa. Kändes som att det blev lite turbaserat sen när man försökte hjälpa varandra.

Social Interaction

Did the fact that each user had information that could help the other user enhance or diminish the experience?

Insåg efter ett tag att det va effektivare att säga till den andra vart delen skulle sätta snarare än att ge dom delen. Blev lite att man va tvungen och vänta på att sätta ut sina egna delar vilket va lite jobbigt.

Could you think of any other features to encourage social interaction between users?

Kan skriva att “blabla” picked up a bone. Eller ljudfeedback att man plockade upp ett ben. Som det va nu va man tvungen att titta själv.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

Mycket titta upp och ner med devicen för att plocka upp grejer. Mindre delar va jobbiga man va tvungen att gå närmare. Hade velat dra delarna för att inte behöva röra telefonen så mycket. Vill också ha större och mer utspridda delar. Kan också ha delarna uppradade för att lättare se vilka man har kvar. Kanske ha en progress bar. Det va väldigt satisfying i hur delarna sattes på plats när de va felvinklade. Vill få någon typ av feedback för att veta när man håller rätt del på rätt plats och kan släppa. Det är tidskrävande att hela tiden behöva släppa och plocka upp men det är klart men kan missbrukas annars såklart.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

-

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

Hade velat dra ben. Kanske gruppa ihop ben, beror på målgruppen hur mycket man borde gruppa ihop ben. Hade velat rotera. Den skulle kunna auto-rotera som hjälpmedel.

C.3 Third Test

Observations:

- En användare ser att den ska vänta på att hosten ska placera skelettet
- Hosten trycker på start för att se om den andra användaren kan se skelettet efter det
- Blev lite problem med tracking och syncning, skelettet spawnade i luften
- Startar om hela rummet
- De verkar lite förvirrade över att skelettet inte kommer fram för den andra med en gång
- Tracking complete!
- Verkar fatta ganska snabbt hur man placerar ben

- Klickar på punkten för att plocka upp ett ben
- En användare undrar hur den vrider på ben
- De verkar inte ha märkt än att man ser varandras hints
- Lite problem med hur nära man ska vara för att den ska fastna
- Ser blåa markeringen, tror att det är för att man är typ för dålig
- Inser att markeringen är för varandra
- Försöker sätta lårbenet men det fastnar inte
- Verkar ha fattat galoppen nu, sätter ut flera ben utan att det verkar vara några problem
- Siktar med pricken, trycker när outlinen visas
- Fick hjälp med att man inte måste trycka på pricken
- "Småbitarna är lite dryga"
- Pratar mycket med varandra
- Rör sig inte runt i rummet
- Pekar i verkligheten var de virtuella objekten är
- Dem tror att den är färdig innan vinnarskärmen visats
- Kan inte greppa sista benet
- De klarar pusslet

Intro questions

Have you used AR applications before?

Användare 1: Typ inte

Användare 2: lite pokemon go men stängde av det

If yes - Have you used many? Can you give examples of which apps you have used?

-

Interaction

How well do you think the interactions used to pick up and place pieces of the skeleton worked? Why? (Depth. . .)

Det var svårt att få tag på de små bitarna. De små fingrarna var jobbigast. Men det kanske är roligt att man får röra sig lite. Coolt att den matchade in på skelettet (djupet). Småbitarna smälte in i bakgrunden (tidningar). Trodde man skulle trycka på pricken för att plocka upp bitar. Missförstånd gjorde att en försökte rotera men lyckades inte. Lite svårt ibland men blev bättre när man rörde på sig.

Feedback

Did you feel that the system gave you enough feedback on other people's actions? When did it? When didn't it?

Hängde inte alltid med. Blev tydligt på de stora bitarna. Den blåa markeringen var lite oklar i början, trodde man skulle sätta sina egna bitar där. Blev bättre när man förstod. Trodde det blåa var en hint för att man var för långsamma först. Kanske hade velat ha en markör för vart den andra tittar någonstans. Blev lite rörigt när man skulle hålla koll både på sig själv och den andre.

Social Interaction

Did the fact that each user had information that could help the other user enhance or diminish the experience?

Det hade varit mycket svårare med fingrar och tår om man inte hade fått hjälp. Blev lite bitter när man försökte sätta ett ben på rätt plats och det inte fastnade. Var nog den enda gången det hände när det hände med lårbenet.

Could you think of any other features to encourage social interaction between users?

Kan göra som vi redan gjort, att en spelare har tillgång till info som den andre behöver.

User Comfort

Did you ever feel discomfort during the experience? Why? How?

I början när det tracka dåligt så var det lite jobbigt men annars kändes det hyfsat naturligt. Lite att benen inte fastnar.

Do you think it would be discomfortable or exhausting to use the application during longer session? If so, how would you avoid this? If not, why?

Mobilen blev varm. Titta upp och ner blir jobbigt efter en lång stund. Kunde ändå göra det mesta stillastående så det gjorde det bättre. Tror samtidigt att det blir mindre jobbigt om det blir mer gå och mindre upp och ner. Jobbigt när man inte fick tag i bitarna.

Accessibility

Did you at any point feel constrained in which interactions you were able to make? Why/why not?

I början trodde en att man var tvungen att rotera benen för att dem skulle sätta sig rätt. Lite oklart hur rätt systemet tyckte det var . Hade kanske velat rotera ben för att se hur dem såg ut och hur bra de passar in.