Master's thesis

2020

Carl Erik Friedemann

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

Master's thesis

NTNU
Norwegian University of Science and Technology

Carl Erik Friedemann

Collaboration And Education With The MagicLeap One

July 2020

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Collaboration And Education With The MagicLeap One

Carl Erik Friedemann

Databases

Submission date:	July 2020
Supervisor:	Simon McCallum
Co-supervisor:	Ekaterina Prasolova-Førland

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Abstract

Virtual, Augmented and Mixed Reality (VR/AR/MR) are related technologies that enable humans to connect with digital environments and with each other. Massive investment in VR/AR/MR from Google, Facebook, Microsoft and others, has created new opportunities and a requirement for research into the effective use of these technologies, especially their social and collaborative aspects. The collaborative aspect of VR/AR/MR is of particular importance following NTNU merger and the corresponding need for supporting educational activities across campuses of NTNU.

This master thesis will focus on AR for collaborative work on 3D content in educational contexts and evaluate AR as a tool for collaboration and learning in different settings. Possible use cases include: anatomy models in medical education; 3D artistic content in art education; urban modelling for architecture students; remote collaboration in a pandemic context. The research questions aim to find the advantages and challenges of AR technology for different type of content; user interface aspects and affordances; the challenges of establishing a shared workspace with AR/MR; maintaining workspace awareness. This work was performed on the basis of MagicLeap technology.

Acknowledgements

I would like to thank my supervisors Simon McCallum and Ekaterina Prasolova-Førland. Additionally, I want to thank Mikhail Fominykh, the rest of the master students at the IMTEL lab at NTNU, and every visitor who helped participate in user tests.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Recent years have seen rapid development of Virtual Reality (VR) and Augmented Reality (AR) technological developments. Last year Valve released a new Head Mounted Display (HMD) called Valve Index, which is able to track the placement of each individual finger on the controller. Magic Leap also launched their MR headset Magic Leap 1 the same year. While HMDs like these are often more aimed towards the gaming industry, virtual and augmented reality has and still is being used for improving communication in teaching and training. As more HMDs become accessible, it is natural to explore more areas in which it could be useful. Very common is the use of VR HMDs in the medical training field. [1]. AR, though popular as a training tool is not as popular of a tool in educational settings, as I will see in section 2.5.

There are multiple needs for collaborative applications of AR which would significantly benefit from the ability to manipulate objects in a virtual space. These apply to 3D inspection and remote collaboration in fields such as manufacture, construction, and maintenance. Similar activities are ongoing with the application of industrial robotics to medicine where the application of 3D visualization tools with multiple simultaneous viewing angles would resolve an issue in that industry.

There exist multiple applications as replacements for video chat in AR, and Magic Leap themselves have released a drawing and sculpting application which can be used collaboratively. However, this application is as of the writing of this paper not available in Norway and therefore not of use. Even if they were available, these have a heavy focus on being used for meetings and calls, rather than creatively and collaboratively. There is also no similar application which utilizes standard 3D modeling tools like the 3D gizmo which will be introduced in section 2.7.1. For this reason, it would be interesting to develop an application with a focus on creative collaboration with such tools.

AR is currently a popular research subject, but as far as I can observe in AR research trends, see section 2.3, there is little to no research on specific technical 3D modeling tools such as the Gizmo which is central to graphical design. Ar is important for enhancing and enriching collaboration, especially in an educational setting. As someone with several years of 3D modeling experience, I wonder what kind of effect the ability to perform more precise 3D transformations in collaborative tasks could have the quality of collaboration. I hope it would increase the quality of collaboration and aim to find out. There is a need for more research in the area of 3D modeling in an educational context.

Additionally, because of the recent Covid-19 pandemic as of the writing of this thesis, remote collaboration has become increasingly sought after due to the lock down nature of pandemics. The implications of the applications go beyond the tools themselves, and extend to significant social value. The research can potentially engage with market needs and market potential as it could impact work processes and business models during a shutdown.

1.1.1 Use Cases

I have looked at multiple use cases in order to discover user needs, some which I discovered were irrelevant while researching, and some which became irrelevant due to Covid-19. Here I will list them and detail why they were or were not kept.

A potential use case would be for art students to convey their ideas for sculptures or other kind of physical art projects through MR. By putting primitive 3D shapes together and placing them in a room, students would easily be able to convey their ideas to others, and demonstrate the collaborative possibilities of MR in the process. There was a meeting held where art students filled out a form whether they would be willing to participate in this, but due to lack of accessibility this would not be followed through.

Several meetings were set up with city planners and architects. However, their interest was in mobile technology and not HMDs. In theory there would be a use case where the 3D manipulation technology could be used in meetings with clients to convey the idea and concept of a proposed architectural project. With their lack of interest however, this was not pursued.

A branch of NTNU architects were contacted, and there were held multiple meetings about use cases where architects would lay out their ideas in a modified reality. This would include placing digital furniture around in a real room and experimenting together. This however would also be discarded as the Covid-19 outbreak limited the number of users who would be able to test using the HMD.

There was discussion with Maori tribe representatives in New Zealand to have a use case based on exchanging culture. In this case 3D models from Norwegian Viking culture and New Zealand Maori culture would be loaded in, and different users could explain their significance to each other and use the models to build a scene. An example would be a mixed culture village focusing on boat culture which is an overlap between the Maori and Vikings. This also fell apart however due to Covid-19.

In order to take advantage of the lock down situation, a final use case was constructed for remote creative and educational cooperation in AR in a pandemic setting. The inability to physically meet with people from outside users' households created a bigger sense of relevance for a remote cooperation application.

1.2 Problem Description

Research the tools, and potential impact of an application which allows for communication through AR in the form of collaborative 3D work. This will be done with the magic leap and include tools to chat, gesture, draw, and manipulate 3D models collaboratively. Through this it should be possible to observe the positive and negative aspects of multiple users communicating through a modified reality.

1.3 Changes due to Covid-19

During the second half of the school year the Covid-19 outbreak spread to Norway which affected the project greatly. On-campus development was banned which limited access to the IMTEL lab, this made testing and development with multiple users difficult as there were only two Magic Leaps which would now have to be spread across this project and another IMTEL project. Only users from the same household would be able to test the HMDs, and this made testing for colocated collaboration difficult as well. Because of this, the project would have to be refocused from colocated to a remote centered groupware. This means that functionality which initially had a high priority such as spatial alignment became less important, while funcitonality like voice chatting became more important. See section 4.4.4 and 4.4.5. Additionally, there were multiple delays due to the general unpredictability of the situation. This affected development and some features to the software would have to be cut to save time.

On a more positive side, this also means that the project could have a higher importance in a time when colocated collaboration is more difficult.

1.4 Research Questions

This section is dedicated to listing the research questions of the thesis. There is one main research question, with 3 sub-questions.

- **Main RQ:** How to support collaborative work on 3D content in an educational setting with AR/MR?
- **RQ1:** How to develop applications on the MagicLeap One?
  - What are the challenges of connecting several MagicLeap Ones?
  - What are the useful additions a MagicLeap One adds to communication?
- **RQ2:** Does the gizmo affect the affordances of 3D manipulation?
  - Would a Gizmo be more useful than a gesture-based grabbing system?
  - What are the advantages and disadvantages of using the Gizmo in this context?
  - How to make the Gizmo as easy to understand as possible in AR/MR?
- **RQ3:** How to support collaborative work on 3D educational content?
  - What are the minimum levels of features needed for a collaborative application
  - What are the affordances of 3D manipulation in single-player vs multiplayer?
  - In what educational contexts is collaborative AR/MR useful?
  - Which industries can benefit from collaborative AR/MR?

1.5 Choice of Hardware

The MagicLeap One head-mounted display was chosen as the tool to develop the end product on. The reason for this was the advantage of processing power the display has over other displays, in addition to the controller. See section 2.4.2 for a more in-depth explanation of the HMD. The controller allows for more precise inputs which would allow for easier handling of the Gizmo. This is important due to the inherent focus the application has on the Gizmo. The choice of an AR HMD as opposed to a VR HMD is due to personal motivation for working with AR as well as AR having the potential to be more effective in teaching than VR, as mentioned in section 2.5.

1.6 Data Generation Method

The intended end users were not readily available throughout most the project and thus most data were generated from questionnaires and short interviews with different visitors at the IMTEL lab. Due to issues with controlling the research environment during visits, the most reliable data during development would be the questionnaires. These included users from backgrounds such as: fellow computer science students, psychology, architecture, art students, geography, and NAV.

Data for evaluation of the final product were generated through semi-structured interviews with different field experts. These experts had experience with game development, user interface design, artificial intelligence, as well as art direction. The interviewees were given a 15-minute video demonstration of the application which they would use as a basis to evaluate the application in the interviews. Four students from Norway, New Zealand and USA were invited to try the application in multiplayer, being three players at a time. The number of users available for testing were greatly reduced due to the lock down taking place. In order to gather more data, a questionnaire with less technologically heavy questions was passed around digitally along with the 15-minute video.

1.7 Research Method

The goal of this thesis was to explore the potential for cooperation on the MagicLeap One by developing a cooperative application. To achieve this I used a combination of qualitative and quantitative research methods. For development, I used the Oates's "Researching Information Systems and Computing" strategy for designing and implementing the application. There are five steps, which are not intended to be a step-by-step guide, but a fluid iterative process. I went through these steps several times throughout:

- Awareness
  This step is meant to gather awareness on a problem to solve.
  In the initial stages I went to interview different focus groups: doctors, artists, and city planners. The doctors said they would be very interested in any kind of project that would allow them to combine 3D models with the physical world, using anatomical dummies with an AR overlay. I tested early versions of the application as well as fully completed creative applications with art students to see their interest, and the students reported being interested in using both AR and VR as an artistic tool. Lastly I interviewed city planners about what kind of application they would want for conveying new building projects etc. An ideal project for them would be something that allows them to place buildings in the street and see it through their phone. Additionally, I studied potentially useful existing products as well as literature on them. In order to have a better understanding of the technical and design aspect, I also

interviewed a VR games expert as well as a designer.

- Suggestion
  In the suggestion step, a concrete idea is formed as a solution to the given problem.
  Through user tests and research on similar applications, a list of requirements was assembled for a general application that could be specialized for multiple fields, see section 4.1
- Development
  Based on the suggestion, some kind of artifact needs to be developed.
  The application went through several iterations of agile development. The first iteration was a very simple proof of concept application which would only be playable for one person. The final iteration was playable for multiple users and would be continued to be developed until delivery. For more details on the development and implementation, see chapter 4.
- Evaluation
  In this step I look closely at the developed artifact and look for holes, whether that be holes in knowledge or shortcomings in a product.
  Final evaluation of the application was done through several interviews with experts from AI, XR and art fields, as well as experiments with multiple users followed by an interview, and questionnaires passed around digitally. See chapter 6.
- Conclusion
  Any results discovered throughout the process are written up and identified. Potential future work to fill gaps are noted as well.
  See chap 8 for the conclusion.

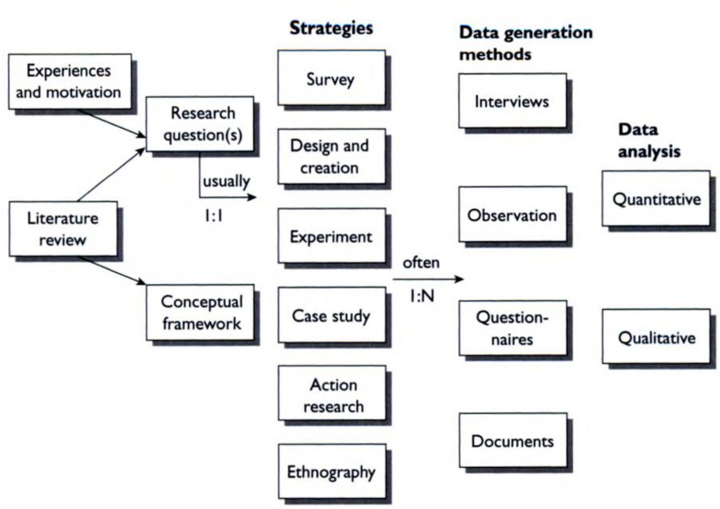

Figure 1: The OATES research method

1.7.1 Early User Testing

Early prototypes would be tested by other members of the VR lab and visitors from different fields. Questionnaires would not be used this early in development as the tests would be performed very often and not necessarily with enough time to fill them out. Instead, the users would be encouraged to think out loud and notes would be taken during testing. The feedback would be taken into consideration and some of it would be built into later prototypes. See 6 for more details on the results of user tests.

1.7.2 User Testing During Development

A video tutorial was prepared for later user tests. Users would watch the tutorials two at a time and test the application two at a time as well. They were given the task to recreate a molecule from an image on an iPad. After performing the task together, they were given a questionnaire to fill out. Users were also encouraged to think out loud and their feedback would be noted during testing.

1.7.3 Final user tests

Due to the Covid-19 outbreak, the ability to perform user testing would be greatly reduced. HMDs, requiring physical placement on a user's face were not feasible since they could not be shared outside of users within the same household. In my case that limited it to two people. The IMTEL lab has access to two ML HMDs. These were separated from the lab due to the lock down and thus colocated cooperation was out of the question.

These experiments would be more structured and played cooperatively and remotely. A fellow student with access to an ML HMD and I introduced the third party to the application. I explained how it works and answered questions while the third user explored the application. Once the user felt more comfortable with the application, the other student would help the third party build a structure cooperatively. The user would then be interviewed for feedback.

2 Background

2.1 Extended Reality

The term Extended Reality (XR) is an umbrella term which refers to the spectrum of VR, AR, and MR. This is also referred to as the "Reality-Virtuality Continuum" illustrated in figure 2. These terms are often mixed and misinterpreted. According to Migram et al. (1994) [2] they are being used without what could be considered as a consistent definition. Hence, it is important to clarify what I mean by VR, MR, and AR. This section will focus on defining and separating the three, as well as provide popular examples.

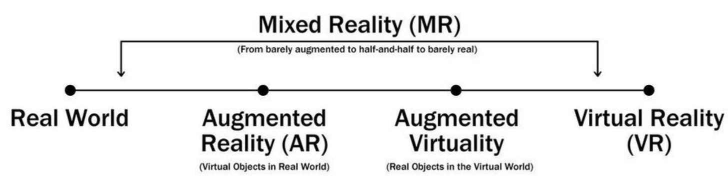

Figure 2: An illustration of the spectrum of XR

2.1.1 Augmented reality vs Mixed Reality

The Interaction Design Foundation defines augmented reality as the real world enhanced by computer generated input, this can mean video, but can also be sound. [3] Most often however, it is done through either video or see-through HMDs. The video-based AR works by superimposing graphics on top of video footage real time, often through a phone camera, with Pokémon GO being a popular recent example. In Pokémon GO the user can view 3D Pokémon standing on surfaces in the real world though their camera phone. [4] There exist several HMDs made for AR such as the Google Glass, the Magic Leap, and the Hololens. Google Glass is marketed towards logistics, manufacturing, and healthcare [5] , and is used by companies like General Electrics (GE). GE uses the Glass during manufacturing to reduce the number of errors by for example displaying whether a screw is too tight or too loose. [6] The Magic Leap has recreational apps such as 'Dr. Grordbort's Invaders' which allows enemies to interact with the geometry of the environment.

Intel defines Mixed reality as something similar in that it also augments reality. However, it also takes input from the physical reality to augment the virtual [7]. To reduce confusion, I will not be using the term MR.

Figure 3: The Pokemon GO app. The pokemon will appear to be sitting on real surfaces.

Figure 4: An example of how the Glass can be used to enhance agricultural work from Googles promotional video

2.1.2 Virtual Reality

Virtual Reality (VR) is the most widely known of the three. The goal of VR is to completely immerse the user into a digital world. [7] It differs from the other two in that it does not make use of the real world, but rather aims to "replace" it. Most recently VR has been used in gaming. Popular examples are HMDs like the HTC Vive, or the Oculus rift. Both HMDs have motion controllers which allow the user to interact with the environment. Though it still falls under the gaming category, it is also used to provide terminally ill patients with the experience of being outside and socializing with other people. [8] [9]

Figure 5: Top: Vive, bottom: magic leap. The Vive has a monitor for each eye, effectively separating the user from reality. The Magic leap uses the see-through display to superimpose the virtuality on top of reality, anchoring it in reality.

2.2 AR Groupware

Groupware is a term used to describe software designed to be experienced collaboratively in a shared digital work space. According to Wang et al. (2006) [10], at least one person needs to be physically next to the augmented task in order for it to classify as AR groupware. Otherwise, it will instead fall under the collaborative teleportation category. The number of users is important as well, as it needs to enable a group of users to perform a task together using the real world.

As Ens et al. (2019) [11] states, most of the research is focused on the underlying mechanical qualities of the applications, rather than the qualities of the user experiences. Although he also states that there is an increase in "more meaningful" investigations on collaborataion rather than the technical challenges on XR. This is due to more of the technical challenges either being lessened or overcome, such as the capacity for replicating physical objects and environments, and network connectivity.

2.2.1 The Six Dimensions

According to Ens et al. (2019) [11], the analysis of AR Groupware can be split into six dimensions. Time, Space, Symmetry, Artificiality, Focus, and Scenario. In this section I will describe the different dimensions in detail.

Time

The time dimension can be divided into two - asynchronous and synchronous. Synchronous meaning collaborative work happens at the same time, and asynchronous meaning the collaborators do not have to work at the same time. The vast majority of research is done on synchronous collaboration.

Space

Groupware can be either colocated or remote. Colocated groupware means all users are located within the same physical space, while remote means at least one of the users are in a separate space. Remote collaboration can give the impression that there is only one user, so proper communication of the other users' presence is essential [10]. The time dimension with the space dimension can be combined to create a so-called traditional CSCW Matrix. Which is often used to classify groupware.

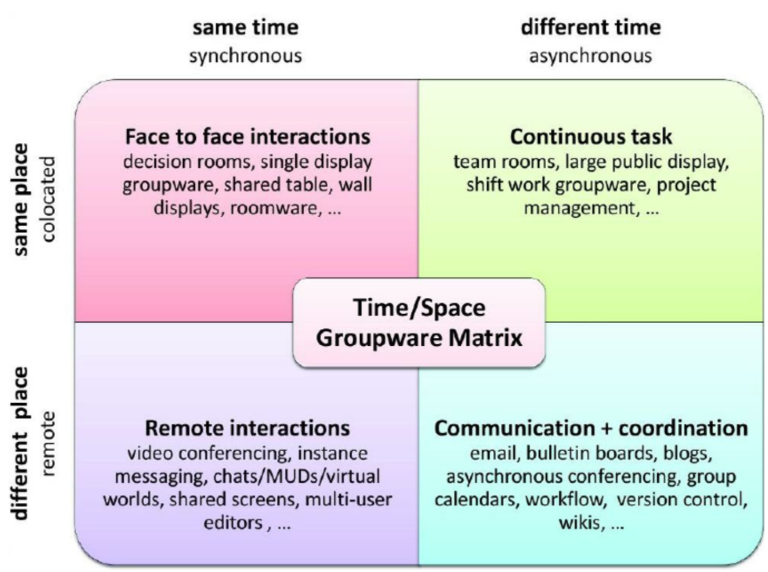

Figure 6: The CSCW Matrix by Johansen 1988

Symmetry

The degree of symmetry is decided by how similar the roles of each user is. If every user has the same role i.e. they have the same permissions and tools, the application is symmetric. If there are multiple roles, then the application is asymmetric. Most the research Ens et al. (2019) [11] examined was symmetric (63%). Of these, most were colocated. Usually in colocated systems the users will work on the same task and thus have the same roles. Remote symmetry exists, but most common according to Ens is asymmetric remote applications. In Remote systems, the users will more often have different tools or tasks. Often the remote user is a field expert, a scenario described in 2.2.1.

Artificiality

With AR and MR there is a degree of artificiality in that it will always involve elements from real life in some way and the question is how much. A mostly physical system will have most of its elements from the physical world with only small augmentations such as annotations or pointers. Mostly digital systems are based more on the digital augmentations in the application. For example in this thesis' software, the focus is on the digital content added to the physical world for context instead of the augmentation adding context to the physical.

Focus

The focus describes the primary target of the collaborative activity. It can be split up into environment, workspace, person, and object. The targets of the collaborative activities in the papers researched by Ens varied. The applications need to be tailored differently based on the focus on the application. An application focusing on the environment will ideally have functionalities enabling the users to explore, or at least see all or parts of the environment of their collaborator. An application focusing on the workspace will include the objects of interest. An object focused application will include a representation of a physical object. Person focused means the application is focused on making the collaborators visible in some form.

Scenario

Most research on groupware is categorized into five scenarios: remote expert, shared workspace, shared experience, telepresence, and co-annotation. Remote expert involves a person with relevant knowledge guiding a local person through a task. Shared workspace is an umbrella term for scenarios where users who collaborate on a task in physical anv virtual workspaces. Shared experiences revolve around the users' personal experience rather than a task. Telepresence revolves around the communication between two or more participants, and finally co-annotation revolves around systems that append virtual annotations on an object or environment.

According to Ens et al. (2019) [11] there has been a large upsurge in research on expert and shared workspace scenarios including a focus on communicating the presence of remote users since 2012. The most explored scenario is shared workspace. Mostly being explored in the context of video games and designing and prototyping.

2.2.2 Workspace Awareness

Workspace Awareness is a big research trend in recent years. This entails knowledge of who is in the workspace, what they are doing, and articulating an understanding of how visual information affects collaboration. According to research from Gutwin et al. (2002) [12], visual cues analogous to real world body language which indicates focus can help provide workspace awareness. Specifically communicating where a user is working and what they are doing is a positive.

Recent developments in technology have made a higher degree of communication in groupware possible, with technology that can closer capture human body language and more specifically gestures. More accurate tracking enables better communication of the hands' and head's position which is then used for gesturing or simply communicating the position of the player. Researchers have found that the use of avatars can help represent the focus of a user as well by displaying where the user is looking and working. These avatars range widely from procedural reconstructions of the user, to pointers and icons, to premade 3D models tracking the user position. Even with these however, it seems gestures like pointing are still difficult in 3D environments. Ens et al. (2019) [11] comments that technology is not yet at the point where I have a framework to communicate all domain- or task-specific gestures and thus developers need to make application-specific avatar designs.

Another important tool for workspace awareness is conversation. Something which affects the quality of conversation is the shared view of the virtual and physical environment. A disjoint view of the environment causes confusion and misunderstandings in conversation. Researchers are still struggling to find general solutions on how to communicate each user's view of the environment, especially with video solutions. In applications where the environment is fixed, usually pointing systems are used, however this too has no good generic solution. [13]

2.3 XR Research Trends

Kangsoo et al. (2018) [14] participated in the second half of a multiple-decade-long research experiment documenting the trends in XR technology. Kangsoo et al. (2018) [14] cites tracking as one of the most popular research topics given the complexity of achieving high quality low-latency tracking. There are multiple techniques for tracking such as Simultaneous Localization and Mapping (SLAM) which uses image recognition to map data in a 3D space. Another is RGB-D Data and Reconstruction which uses depth cameras to procedurally create 3D models of the environment more densely and efficiently than SLAM. This technique is used in an example later. Lastly are hybrid methods which use multiple techniques simultaneously such as GPS location and SLAM.

Additionally and more relevant to this thesis are the interaction techniques documented. First off is the Tangible User Interface (TUI) which integrates a physical object into the user interface such as a box or a piece of paper with print on it. The opposite of this is an Intangible User Interface (IUI) which makes use of floating mid-air interface elements. Kangsoo et al. (2018) [14] compares it to popular science fiction UI such as the UIs depicted in popular movies like Iron Man. According to Kangsoo et al. (2018) [14] they are considered too tiresome in some domains, but are often used in gaming and rehabilitation programs.

Kangsoo et al. (2018) [14] quotes Zhou et al. (2008) [15] on three different areas of limitations with current AR systems: Problems with using physical objects and gestures, lack of good human factors design, and poor interaction design. Gesture tracking has improved greatly in the recent decade, but TUI elements are still troublesome. The physical difficulties of using AR have been reduced with the rise of handheld AR, but there is still room for improvement with cognitive interaction design. Most of the research documented is on handheld AR and tracking.

When listing future work, Kangsoo et al. (2018) [14] remarks on a lack of research on collaborative systems. There is also a lack of research on the social aspects of AR. He also mentions Spatial AR as an example of a relevant application to research, which will be described in more detail later.

In both the Zhou et al. (2008) [15] and Kangsoo et al. (2018) [14] papers on AR research trends, there is a lot of documentation on the technical side of interaction techniques i.e. which kind of UI, but little documentation of the interaction controls, i.e. how to select an element in the UI. For example there is a mention of the quality of gesture tracking but no mention of the difference between controller based interaction and gesture based interaction. More specifically there is a need for evaluation of different techniques for manipulating an element in the application.

2.4 Head Mounted Displays

Head mounted displays are digital displays which can be mounted directly on the head, often in the shape of a headset in combination of glasses. This separates them from Helmet Mounted Displays which are mounted on and often built into the helmet. HMDs are designed such that a display screen will be in front of the user no matter their position. This allows for an immersive experience. [16] Most are made with an individual display for each eye. These displays can be used to augment or add something to my surroundings, or replace them entirely. (see 2.1) Often the displays will be tethered to a computer which will do most of the processing. Some come with their own lightweight computer which can be carried by the user, such as the HP VR Backpack, or the Magic Leap. [17] There are also displays which are completely standalone, such as the Oculus Go or Oculus Quest.

2.4.1 History of Head Mounted Displays

Up until at least the 90s, head mounted displays used to have issues with tracking. which in turn created issues with creating seemingly correct stereoscopic images. This means as the head moved around, the visuals on the display would drift because of the inability to keep track of where the head is pointed. In the cases where the tracking was done correctly, it would still suffer from delays as the technology was not sufficient at the time. [18] [19] In 2012, Palmer Luckey introduced the Oculus Rift. This would eventually bring a lot of attention to HMD technology as Facebook bought it for $2 billion and resulted in a large growth in the industry. [19] In 2016 the Microsoft Hololens was released, which was regarded in 2015 as "the only holographic computer". [20] This would set an important milestone for AR as it would be one of the few consumer AR products not based on gaming. [21]

2.4.2 MagicLeap One

The Magic Leap One is a head-mounted virtual retinal display developed by Magic Leap, inc. and officially released and distributed in August 2018. It consists of three parts, the headset "lightwear", a motion controller, and a computer tethered to the headset called "lightpack". The Creator Fact Sheet describes it as "A lightweight, wearable computer that enriches your experience in the real world with digital content." [22]

The display on the headset is see-through, allowing the headset to superimpose images on the

Figure 7: The Magic Leap developer kit.

user's view of the physical world. It uses multiple sensors to 3D scan the environment which makes the headset aware of its 3D position and environment, thus being able to maintain the position of digital objects in relation to the physical world. MagicLeap refers to this as "headpose". When the MagicLeap One loses headpose, it loses track of the position in the physical world. The software running on the headset can also use the scanned environment, meaning they can be contextually aware, and can recognize rooms it has been in before [23]. This opens possibilities for 3D objects to collide with real world objects, and the sound design can make use of the geometry of the physical world as well. The founder of MagicLeap claims that this kind of technology will replace screens in the future [24] [25].

The headset also tracks the position of the motion controller, which offers six degrees of freedom, referred to as 6DoF. This means it keeps track of forward, backward, up, down in real time. It also keeps track of rotations such as roll, yaw, and pitch. It has two buttons: the home button and the bumper. There is a pressure sensitive trigger below the bumper, and a touchpad [26]. See figure 8 for the placement of these buttons.

The field of view on the MagicLeap One is smaller than most VR HMDs which currently have horizontal field of views up to 110 horizontal degrees [27], but bigger than most AR HMDs. The MagicLeap One has a 40 degree horizontal field of view, and a 30 degree vertical field of view [28]. This makes the field of view 45% larger than that of the Hololens1. The human field of view is approximated to be about 220 horizontal degrees, so despite being arguably the largest in the field, users will still experience the field of view as fairly small. [29]

Additional to the constraint of the field of view, is what the MagicLeap team describes as the

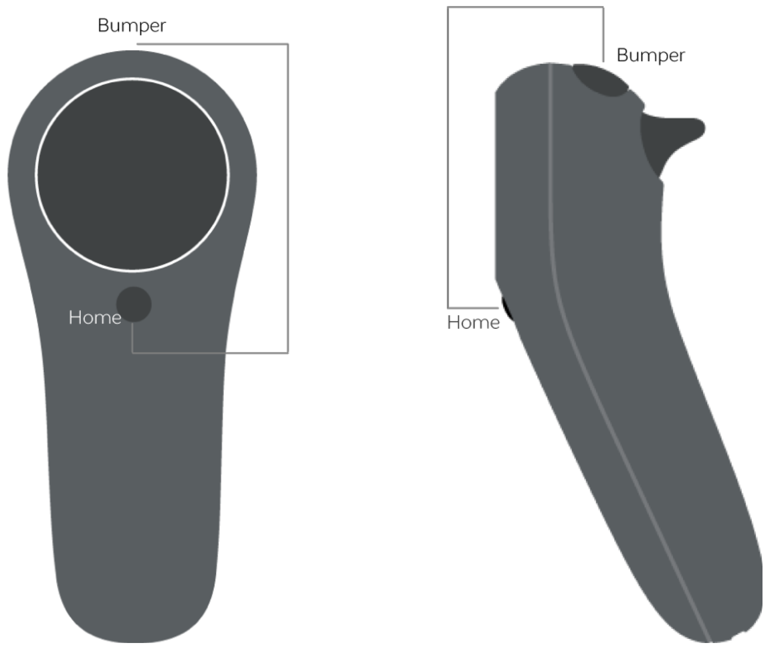

Figure 8: The Magic Leap motion controller

"view frustum", which in development is referred to as the "clipping plane". The view frustrum is about 37 cm long, which means that anything less than 37 cm away from the HMD will be cut off. This is something which at the moment developers will not be able to change as it is "handled by the device" according to the MagicLeap development team. [28].

2.4.3 Hololens (1st generation)

The Hololens it the "world's first fully untethered holographic computer" according to Microsoft, the owner of the Hololens [30]. Like the MagicLeap One, it is a head-mounted virtual retinal display. Unlike the MagicLeap One however, it is not tethered to a separate computer. The main interaction method with the Hololens is hand gestures. It does to some extent have a controller, meant to be an alternative to hand gestures [31]. The "Hololens Clicker" is a small device which can be fastened around the finger and flicked, as opposed to the larger hand movements needed without it. There is also a button on it as an additional interaction method.

The Hololens HMD has many of the same, or similar hardware specifics as the MagicLeap One. It has see-through displays referred to as "waveguides" allowing it to superimpose images on the physical world much like the MagicLeap One. There are also four "environment understanding cameras", a depth camera, and an inertial measurement unit, which allow the Hololens to scan the environment and keep track of its position and rotation in space. [30]

The Hololens has a horizontal field of view of 30 degreesm and a vertical field of view of 17.5 degrees [29]. This, as well as the lack of a sophisticated controller was the reason for not choosing

Figure 9: The Hololens HMD.

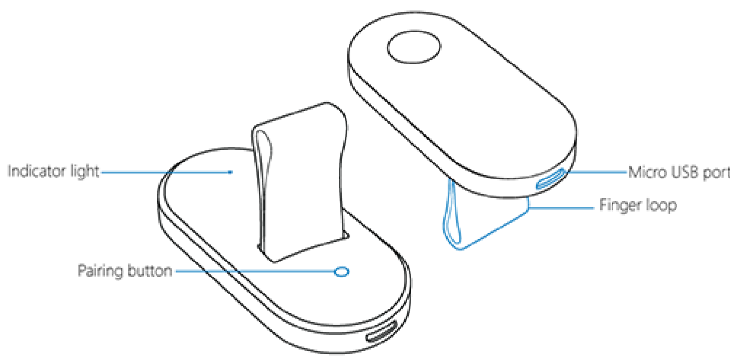

Figure 10: The Hololens Clicker.

the Hololens for development.

2.5 AR and Learning

Augmented Reality has been found to have potential to improve student learning [32], although there seems to be disagreement around the actual extent of the usefulness and effectiveness in different areas, according to Radu et al. (2014) [33].

Sommerauer et al. (2018) [34] conducted a paper survey on the current AR theoretical and empirical foundations. On the data collected, none of them reported AR implementations without positive results. The trends they found in different implementations involved the following: spatial contiguity, in which objects are aligned with digital counterparts, signaling which uses trigger images to create associations, and multimedia to reduce cognitive loads for example by playing spoken words instead of displaying them written. Mobile AR solutions using maps are often used as well, which encourages users to move around for different purposes.

In studies where there had been real life collaborative tasks combined with AR, communication between participants was labeled critical, and several of the experiments implemented the same voice communication systems often used in video games. However, none of the research experiments used any standardized communication systems. They all implemented their own instead [34].

Radu et al. (2014) [33] conducted a research experiment surveying papers and finding multiple benefits as well as detriments to using AR in an educational setting. These are as following:

Learning Benefits from AR

- **Increased content understanding** Compared to books, videos, or PC desktop experiences; students were more easily able to understand content through AR.
- **More spatial understanding:** reserach surveys found that students more easily understand spatial and structural relations in AR vs PC desktop experiences and books. There was also research showing students were better able to transfer knowledge to real life situations with AR rather than VR experiences.
- **Learning language associations:** One study found teaching Chinese characters with AR to be more effective than books, while another found low and average achievers learning more than high achievers with AR.
- **Long term memory retention:** Research found that knowledge gained through AR was less likely to be forgotten a week later, as opposed to knowledge learned through books or video media.
- **Improved physical task performance:** During training or performing physical tasks, users can perform faster, with higher accuracy and precision with AR guidance.
- **Improved collaboration:** Research has found that the degree of collaboration is in some cases higher when AR is involved in the collaborative task.
- **Increased student motivation:** The use of AR increasing the users' enthusiasm to engage with AR is noted through several of the surveyed papers. They reported having more fun

and being more willing to repeat the experience. Applications which have a lower degree of usability are still often reported to be more fun and satisfying than the non-AR applications.

Learning detriments from AR

- **Attention tunneling:** Multiple papers found that AR applications will sometimes require more attention from users, causing them to forget critical elements of a task. Some users would be too focused on the AR experience and forget their environments. It was also reported that AR headsets can cause "tunnel vision" likely due to a low field of view.
- **Usability difficulties:** Despite users overall liking the AR applications more than non-AR, most applications in the reserach papers were rated as more difficult and less usable than their non-AR counterparts.
- **Ineffective classroom integration:** One paper found their implementation of AR in the classroom to negatively affect the classroom experience by limiting the engagement with educational content. The non-AR solution had a higher degree of student engagement and discussion.
- **Learner differences:** Some research papers showed that AR is not always suitable for both low and high achievers in the classroom. It might need to be specifically tailored to the capabilities of the audience.

It is clear to see that AR, has effects on learning which can be both positive and negative, but the most relevant piece of information to extract from this is the observed increase in collaboration. The surveyed papers found that users preferred to use their bodies to interact and manipulate content. This contributed in motivating users to learn and overcome difficulties of using the application. Users were more excited to use difficult AR systems than they were with easy to use non-AR systems. Factors which affected enthusiasm involved attractive graphics, availability of clear feedback, internal goals, and balance between challenge and personal skills [33]. Sommerauer et al. (2018) [34] also found that implementing design choices from video games can result in a higher enthusiasm for the applications. Elements like storytelling, mini-games, leaderboards, badges and points.

One thing we can see while going through AR and learning trends is that there is little support or research on rich cooperation with 3D content. This could not only be relevant for education but many other settings as well.

2.6 Network Choice

A lot of network coding is required to make a multiplayer application from scratch. Instead of doing this, I decided to use an already existing multiplayer API. These were the available choices I found:

1. UNet:
   UNet is Unity's local multiplayer API. It is integrated into Unity and includes a network manager and multiplayer-aware scripts. It is also deprecated and will be replaced with a different system in the future. [35]

2. Photon:
   Photon is a multiplayer API which is also tightly integrated with Unity. It has a very active community and developers which often add to the discussion in forums. They have a free 20 concurrent users plan [36].
3. Firebase:
   Firebase is a database system which comes with an SDK that can be used for handling of multiplayer applications in Unity [37].

Of these APIs, I chose Photon, because it is not deprecated like UNet, and it is very specifically developed for Unity, as well as a popular choice for Unity. Many other developers at the IMTEL lab have also used Photon which made it easier to ask for help in development.

2.7 3D Editing Software

There are multiple ways to create 3D digital content. One way to do it is to make a physical, real sculpture, 3D scan it, and then edit it with 3D editing software. It can also be created digitally from scratch. Regardless, at some point it will be placed in a 3D editing software which is displayed on a 2D screen. There exists software which will allow you to edit 3D content in 3D space with VR HMDs 3.2.2 3.2.1, but this is not the industry standard. This chapter will explain different 3D editing software. There are many aspects to 3D graphics software, but for this thesis, the most important details are the affordances the programs have in order to allow the user to import, edit, and export 3D geometry. I will examine different desktop 3D applications to see their UI trends with Gizmos, selection and general manipulation.

Industry 3D desktop programs are all similar in that they allow the user to import, add and edit 3D geometric shapes, export them, etc. They have similar user interfaces, and similar tools for editing 3D geometry. As will be discussed below, most of this is done with a multitude of UI buttons, draggable axes, and a reliance on keyboard shortcuts. Importing and exporting is done similarly to other editing programs. Pressing File, Export, and choosing a format. The specifics of 3D camera navigation are not necessarily relevant, seeing as keyboard shortcuts can depend on the settings. However, it usually works by moving the mouse in combination with holding down one or more keys. This will for example pan the camera, zoom it in or out, move it to the sides, etc. This is different from VR and AR applications, as the camera in those cases would likely be mapped directly to the headset.

Mapping 3D space onto a 2D screen will of course remove one dimension, but the user still needs to edit in 3D space. The computer mouse also only moves in 2D, so there is no way to work 1:1 in 3D. This means the program must provide tools which have the affordance of editing in 3D space. Often these are referred to as 3D manipulators, or gizmos. The gizmos themselves look similar from program to program, as the design has become an industry standard. See 2.8.2 for a comparison of the different gizmos.

Selection is done similarly in each program. Clicking on an object with the cursor in the 3D viewport, or on an outliner, will select it. Unless specified, this is how it is done in each of these programs.

2.7.1 Gizmo

The Oxford Learner's Dictionary defines a Gizmo as "a general word for a small piece of equipment, often one that does something in a new and clever way" [38]. According to Autodesk, 3D Gizmos are tools which help the user scale, rotate, and move something [39].

When editing in 3D, there needs to be some way to manipulate the geometry. Often the tool used is referred to as a 3D manipulator, or a Gizmo. In this thesis, the word "gizmo" will refer to the 3D manipulator, meaning a UI element which allows to easier rotate, scale, and translate objects precisely. It is always represented in 3D space, and thus separate from other UI flat elements such as text fields, or buttons. Gizmo designs differ between programs, and can in theory look like anything, however there are design traits which often repeat.

The instances of Gizmos we will look at will usually be placed at the pivot point of the current operation. The Gizmo has three different ways to transform geometry. Rotating, scaling, and translating. Rotation is often portrayed with arcs or circles. One for each axis. Scaling replaces the arcs with lines with cubes at each end. Translation is portrayed with arrows. The tool is used by clicking and dragging a specific part of the gizmo. The affordance here is that each part of the gizmo will manipulate the corresponding axis. Often there will be extra parts on the gizmo which allow the user to perform transformations on multiple axes at once. This gives the user the ability to apply changes to several locations, scale, and rotation attributes at once, as opposed to editing them one by one in a text field. For the case of this thesis, studying the gizmo is beneficial as the user will be without a keyboard, and editing coordinates will be cumbersome. A more "hands on" tool like the gizmo could be better for precise transformations.

2.7.2 3DS Max

3Ds Max is a 3D computer graphics program developed by Autodesk. It was initially released in 1996 and is currently used by various companies in industries such as Aerospace & Defense, Artistic & Photography Services, and Advertising, Marketing & Public Relations [40]

Selecting a 3D model with the cursor will highlight it with an outline and display whichever type of gizmo the user has activated. The gizmo will appear at the "center" of the selected object. The definition of center is dependent on the current settings which can be changed by a dropdown menu in the interface. It can be either Pivot Point Center which uses each individual pivot point center of the selected objects, Selection Center which calculates the geometric center between the selected objects, or Transform Coordinate Center which uses the coordinate center as a pivot point [41].

2.7.3 Blender

Blender is an open source 3D computer graphics program developed by Blender Foundation initially released in 1998. Initially being used by hobbyists, it has become more mainstream in recent years and several companies have switched to Blender from subscription-based programs. Blender is used in industries such as advertising, marketing & public Relations, sporting goods, and golleges

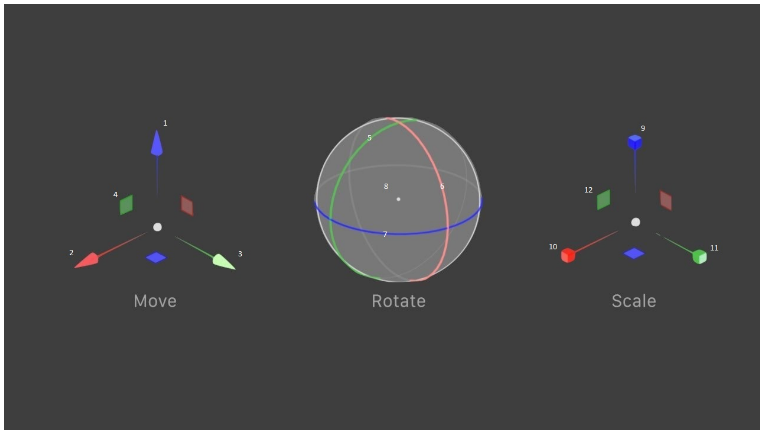

Figure 11: This is an example of a Gizmo from Blender as of 2.8. In Move mode, dragging on 1 would drag the selected object along the Z-axis. Dragging on 2 would drag the object along the X-axis. Dragging 4 would drag the object along the X- and Z-axis. In rotation mode, dragging on 5 would rotate the object around the Y-axis, while dragging on 8 would rotate the object on all three axis. Scaling works the same way as moving, except it would scale on each axis instead of move.

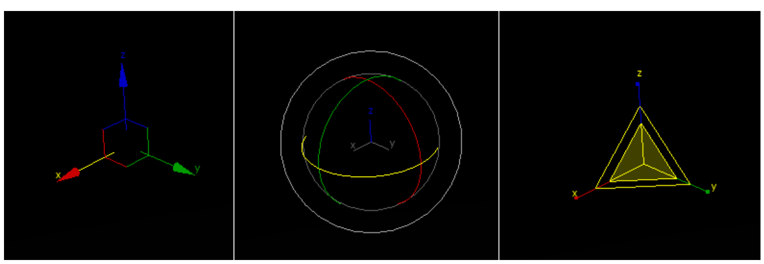

Figure 12: The Gizmo as seen in 3DS MAX. It functions very similarly to the Gizmo in Blender 2.8. One difference which is not immediately obvious is the triangle shapes in scale mode. Selecting the inner-most triangle will activate uniform scaling, while selecting the outer triangle will activate scaling on the selected plane.

& universities [42]. [43]

Like 3DS Max, selecting a model with the cursor will highlight it. As of Blender 2.8 the gizmo is hidden by default and must be enabled. Also, like 3DS Max, the gizmo will appear in the center, which again can be changed by a dropdown menu. Compared to 3DS Max, there are a couple more options for the center. Active Element selects the center of the most recently selected element, Median Point calculates the center point between the centers of the selected objects, Individual Origins uses the center of every object, 3D Cursor uses a separate marker placed by the user called 3D cursor, and Bounding Box Center calculates the center of the bounding box of the selected objects. The user can also change which axes the gizmo is aligned with global, local, gimbal, normal, screen, and 3D cursor.

2.7.4 Maya

Maya is a is an industry standard animation tool owned by Autodesk and was initially released in 1998. Although mainly used for 3D animation, it can also be used for 3D modeling. It is used in a multitude of different industries like movies, TV, games, advertisement, etc.

Selecting a model in Maya will highlight it, but not in the same way as the previous programs. The selected object will have all edges highlighted, making topology immediately visible. This is useful when doing operations such as adding loop cuts, because it allows the user to immediately see how the geometry of the model has changed during the operation.

The gizmo exists in Maya in form of the Universal Manipulator Tool and the combined Manipulator tool. Both combine the scale, rotate and translate tools into one, like with most other programs. The difference between the Universal and the Combined Manipulator is that the Universal does not work on components like vertices, faces, etc. There is also a visual distinction between the two. Rotation, scaling and translating is done by clicking one of the axes and dragging it, or entering a number in the box that appears. Clicking in the center will also allow the user to move it around freely. Settings like snapping, pivot points and transform space can also be changed through the UI. The pivot point can be either the center of the selected object or moved somewhere else by clicking and dragging. The transform space and be local or global.

2.7.5 Sketchup

Sketchup is an industry standard tool initially developed by @Last Software in 2000 for architectural models landscape design, mechanical engineering, etc. It is less of a creative 3D modeling software and more of a tool to create blueprints with mathematical formulas and measurements. The UI is less based on visual cues and it relies more on input of specific numbers than cursor movement like the others, though adjusting measurements with the cursor is possible.

Selection in Sketchup works a little differently in that the user does not select objects but rather faces, edges, and vertices. Selecting either of these will highlight them blue and selecting an entire object will therefore highlight everything with blue. The traditional gizmo does not exist in Sketchup, instead there are completely separate tools for scaling, rotating and translating. Translating is done by selecting the Move tool and clicking on the object to move, which will then follow the

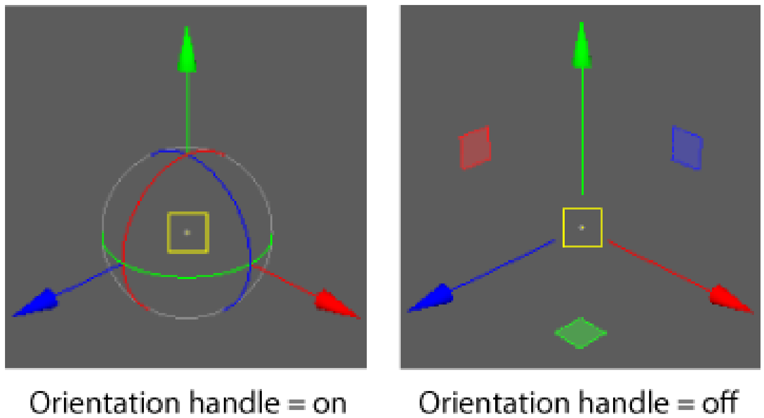

Figure 13: The Combined Manipulator as seen in Maya. It functions very similarly to the Gizmo in Blender 2.8.

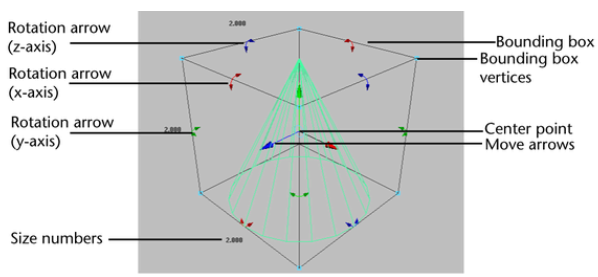

Figure 14: The Universal Manipulator in Maya. It is slightly different visually from the Combined Manipulator, and includes a bounding box.

cursor until the user clicks again. Rotation is done by selecting an object and selecting the Rotate tool, defining a pivot point and moving the cursor to rotate. Finally, scaling is done by using the Measure tool and changing the length of a specified edge on the model. This will scale the whole model or parts of it proportionally depending on the settings.

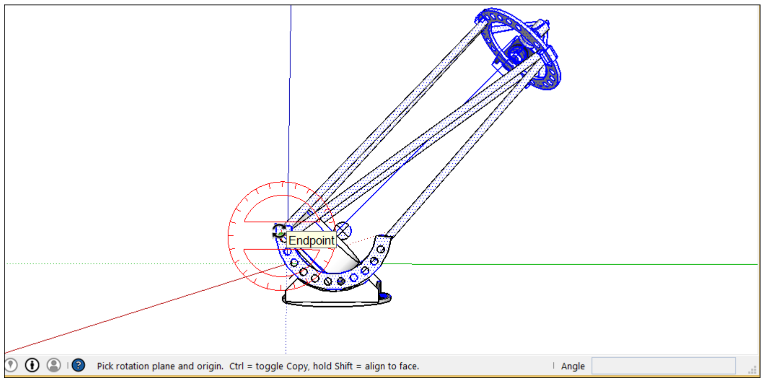

Figure 15: A screenshot of the rotate tool in Sketchup. The circular ruler indicates the pivot point of the rotation and can be turned by dragging with the cursor or inputting a number in a context menu that pops up at the bottom of the screen.

2.7.6 Zbrush

Zbrush is an industry standard 3D sculpting tool used in the movie and video game industry. It was initially released in 1999 and is produced by Pixologic. It is a desktop application built for use with a keyboard and tablet pen or mouse.

Selection is different in Zbrush in that the concept of having selected an object is removed in a sense. Instead there is only the active object, and the rest of the objects are permanently static. The reason for this is that ZBrush has a 3D editor and a 2.5D canvas editor, where anything not selected will become like "paint" or "a pixol" as ZBrush calls it on the 2.5D canvas.[44] Objects in Zbrush are called ZTools, which are located on a list in the UI. Adding a ZTool to the 3D space (referred to as a scene in ZBrush) will activate it and signal this by giving it a brighter color than other ZTools in the scene. The user can scale, rotate, or translate the ZTool by activating said tools from buttons in the UI. Otherwise any other mouse click on the scene will add another ZTool and render the previous ZTool unselectable permanently. Enabling Edit Mode before clicking on the scene again will signal that this is the model currently worked on. Exiting edit mode will deactivate the ZTool and thus make it unselectable. Another ZTool needs to be painted onto the scene in order to select it for editing. ZTools have subtools which are separate meshes within the ZTool which can be modified in edit mode. They have to be selected either by using a hotkey or by selecting them from the list. Editing the subtools in the ZTool will update all instances of the ZTool and so no progress is lost when exiting edit mode and deactivating it. [45]

There are multiple versions of the Gizmo in Zbrush, the Transpose Line and the Gizmo 3D. The

Gizmo is the default when activating Move, Rotate or Scale mode. It will adjust its orientation based on the polygon being clicked on with the cursor. It will also adjust the orientation based on the direction of a click and drag operation. To reset the Gizmo so it aligns with the global XYZ axis, the user can click Mesh to Axis and Reset Mesh Orientation operator while in Unlock Mode. All these are floating UI buttons attached to the Gizmo. Some interesting differences between the Zbrush Gizmo and other Gizmos are the different options for Gizmo positioning when transforming. There is a sticky mode which will snap the gizmo back to the original position after releasing the mouse, and a lock mode which will have the gizmo stand still while transforming. Other differences are the lack of options for changing pivot points. The Transpose Line acts as a replacement for pivot point options, as it is similar to the Gizmo except it draws a line between two points. Each of these points act as their own Gizmo which pivots around the other. The user can also automatically position the pivot on the selection through a UI button which follows the Transpose tool.

2.7.7 Unity

Unity is a popular video game engine which is used by both professional and hobbyists. It separates itself from the previous programs on this list by not being an animation tool or a 3D modeling tool, but it does share a lot of similarities in UI.

Objects can be selected either with the outliner in the UI or by clicking on it in the scene. Selected objects are highlighted with a bright orange outline. The Gizmo as well works similarly to other programs on this list. Like the others, it has three different modes: transform, scale, rotate. The user can transform on a given axis by clicking and dragging on it similarly to the previously described programs. Clicking in the middle of the gizmo will transform freely on all axes in the given transformation mode.

2.8 Comparing 3D graphics desktop programs

Now that I have described the different main desktop 3D programs available, I will compare them to see key differences and similarities and clarify which trends would be relevant to the application I develop.

2.8.1 Selection

It is clear through these programs that highlighting a selected object by giving it a colored outline or silhouette is the industry standard. The biggest difference between them is the color choice for the outline. For all of them except ZBrush, the color of the outline is the same regardless of the color of the object. ZBrush is the only software that stands out in this context where it does not have any outline but rather dims the color of every non-selected object. Despite ZBrush being a very popular choice for 3D development, it does not follow convention and is therefore dismissible in this context.

The way selection works is also nearly the same in every program, clicking on any object in 3D space or the reference to the object in the outliner in the UI sets the selection to that object. The difference here is in ZBrush where deselecting an object in 3D space will permanently remove the

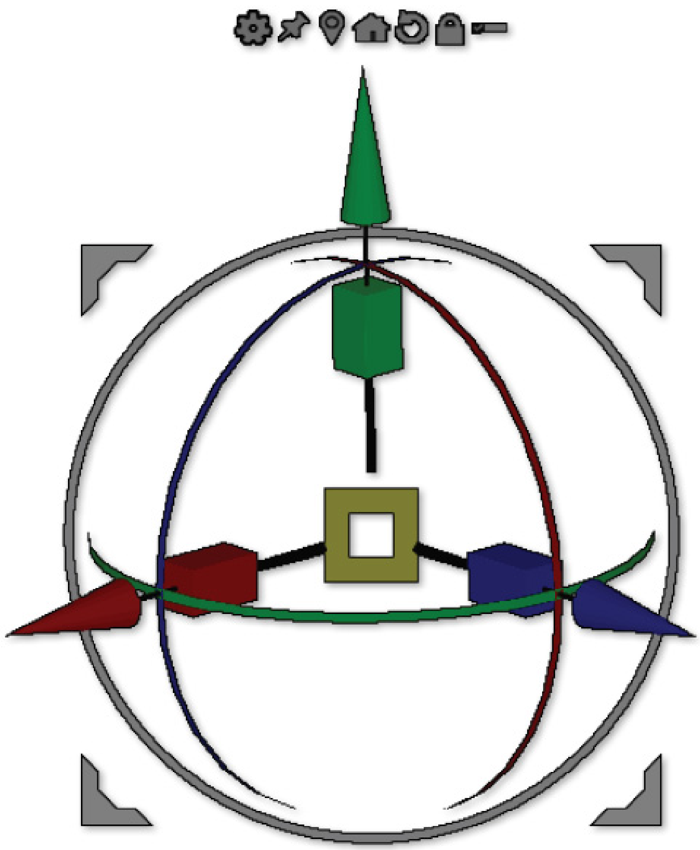

Figure 16: The standard Gizmo in Zbrush displaying all three modes at once: transform, scale and rotate. Above it there are floating UI buttons which control the snapping behavior of the gizmo as well as resetting it.

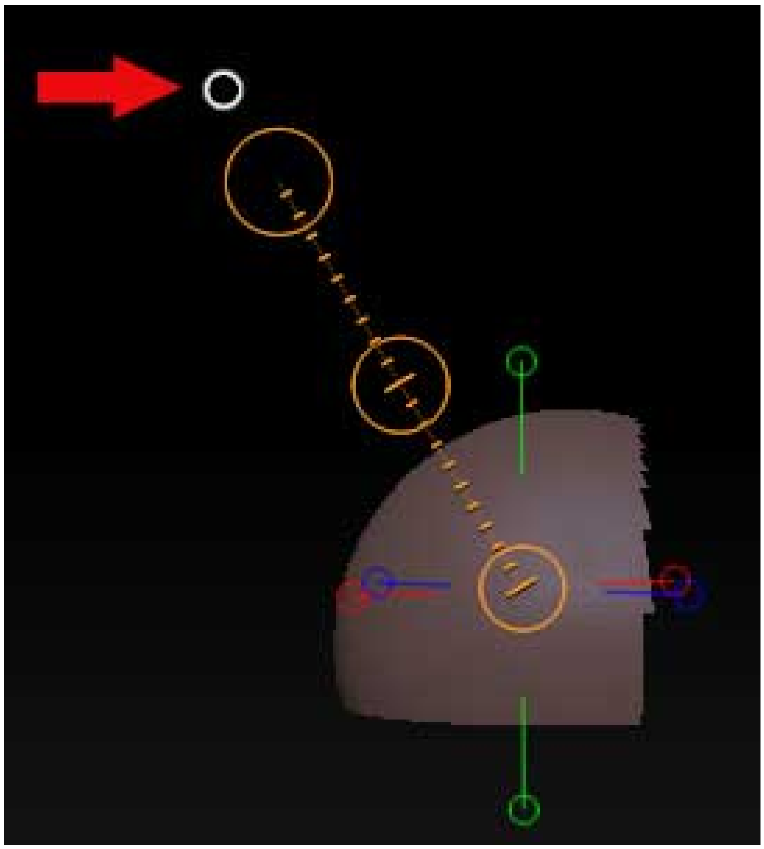

Figure 17: The ZBrush Transpose Tool. The arrow points to the floating UI button which sets the pivot point to the selection.

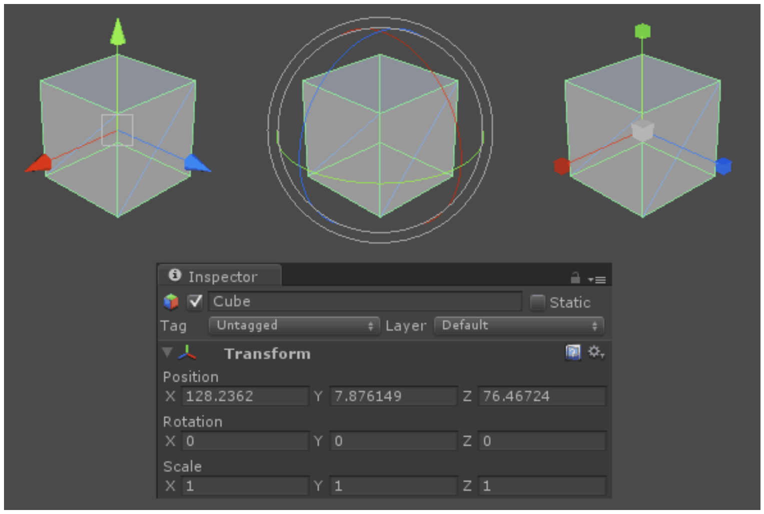

Figure 18: The Unity gizmo. From left to right: Transform mode, Rotate mode, Scale mode.

user's ability to interact with that instance, and it will have to be selected in the ZTool list again. This is so significantly different from the way other 3D graphics programs work and so specific to Zbrush's "2.5D canvas" that it can be dismissed.

2.8.2 Gizmo

Although there are slight variations in design for gizmos throughout these programs, they are mostly similar in design. The translation mode puts arrows on each axis, the scale mode has boxes on each axis, and the rotate mode uses arcs. Each axis has its own color. Usually X is red, Y is green, anz Z is blue, although the Y and Z axes are sometimes swapped between programs. The biggest differences here are with Sketchup and again ZBrush. Sketchup does not have a traditional Gizmo and so the scale, rotate and transform tools all look different from the other gizmos. ZBrush has a traditional Gizmo as well as a non-traditional Transpose Tool. The Transpose Tool and Sketchup's array of transform tools are different enough to be dismissed, as those design choices are exclusive to the software. It is safe to say that a multi-colored gizmo with arrows, boxes and arcs is the industry standard.

2.9 Suggested Controller Scheme

The magic leap developers page has a section about best practices for UI with the Magic Leap. The page can be read here: https://developer.magicleap.com/en-us/learn/guides/design-manipulation

In this section I will go through their suggestions, and what I decided to keep as well as what I decided to discard in order to make space for more functionality. Several changes were made due to feedback in user tests.

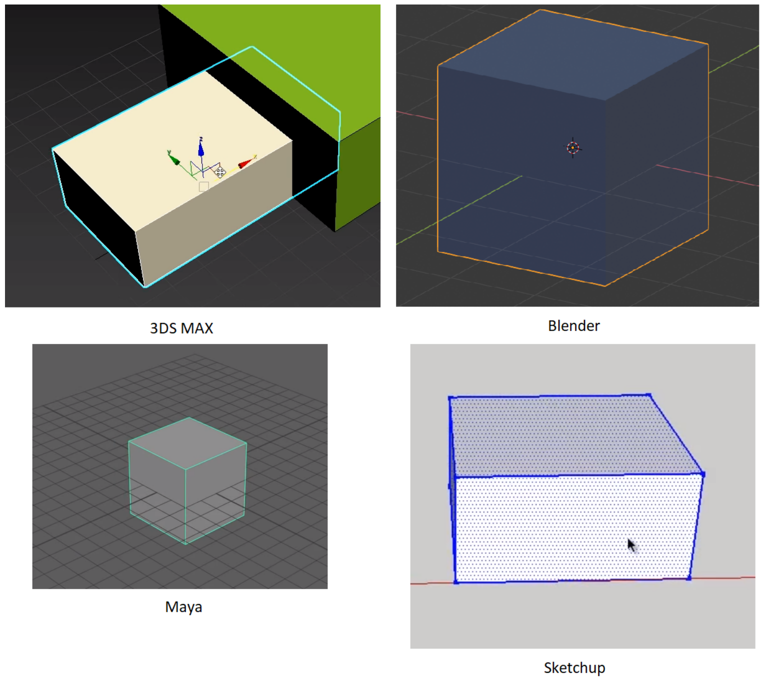

Figure 19: What selected objects look like in 3DSMax, Blender, Maya, and Sketchup.

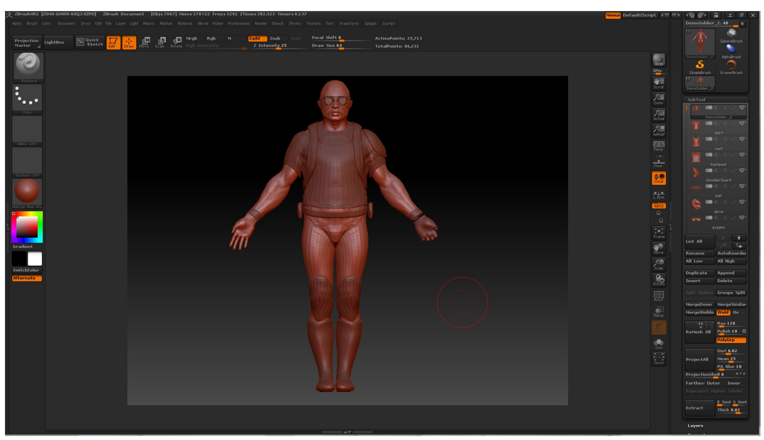

Figure 20: An example of a selected submesh in ZBrush. In this example the body is selected, as can be seen by the brighter color than the other submeshes, as well as it being highlighted in the list of submeshes on the right.

2.9.1 Manipulation controls

On the MagicLeap developer website there is a subsection dedicated to 3D manipulation which details the preferred control scheme for selection, rotation, scaling, and moving.

They suggest that users need to be able to scale, rotate and move around objects depending on what kind of object they are. The developers suggest that UI interfaces need to always face the camera, and thus not be as manipulable, while other "in-game" objects often need to be at least rotated and moved. They then go through each form of transformation and detail their suggestion for how it should be implemented controller-wise.

Rotation is suggested to be implemented with the touch pad. A circular swiping motion should rotate the object in the direction of swiping.

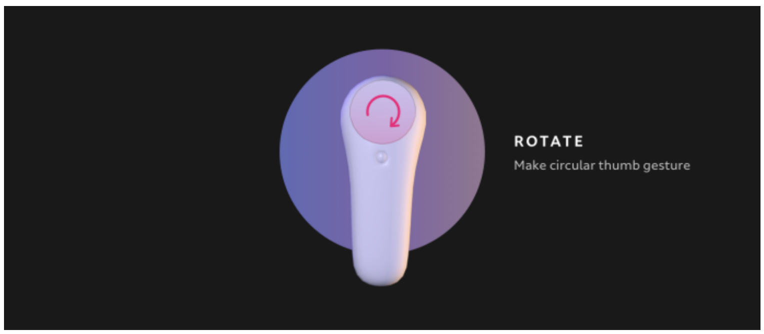

Figure 21: Rotation as suggested by the MagicLeap dev team. The circular swiping motion indicates the direction of rotation as well as the amount.

The Magic Leap developers suggest implementing scaling through the touchpad as well. They recommend scaling by touching the touchpad on the left and right side. Touching on the left would scale the object down, while touching it on the right would scale it up. What this could mean is that as the user holds their thumb on the touchpad, the object would scale continuously. It could also mean that tapping the touchpad would incrementally scale the object each time the user tapped, and that holding the thumb down would not do anything beyond the initial tap.

Thirdly, the developers suggest adding a "push and pull" functionality to the touchpad. Touching the bottom part of the touchpad would pull the object towards the user, while pushing the top would push the object away.

Lastly, there is a section about "grab and place", in which the developers suggest a highlighted object should be grabbed and held when holding down the trigger.

2.9.2 Content Selection

In any application there needs to be a way to select and the Magic Leap developers wrote their best practices for this as well. This includes selecting objects in the application and menu navigation which requires selecting items and buttons.

The first method of selection they suggest is with a ray, which is used by aiming at an object.

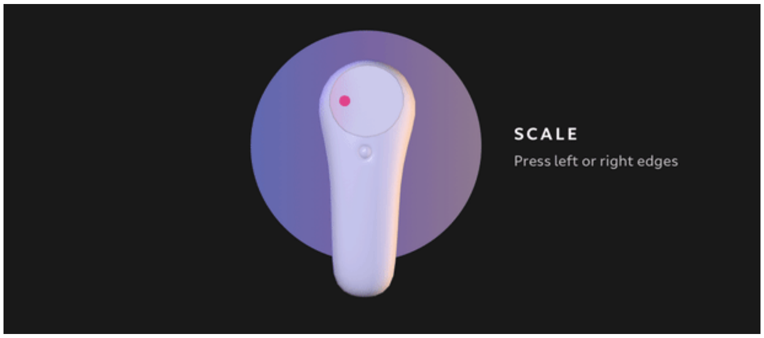

Figure 22: The MagicLeap suggestion for how scaling would work. In this illustration, the user is tapping the left side which would reduce the scale of the selection.

The object then becomes the focus. This is labeled as "direct targeting". The second method they suggest is "indirect targeting" which involves use of the touchpad. This does not require the user to point their controller at anything. Both are suggested to be combined with the trigger to perform a selection, and with a tap on the touchpad as a form of redundancy. They mention that redundancy is important as it makes the application more accessible. The MagicLeap development team also suggest that these selection tools be usable universally throughout the app. This means that if it is possible to select objects with a ray, this ray should also work on buttons in menus, instead of having to switch to for example the touchpad as a tool for selection. They suggest that being forced to switch can be frustrating.

2.9.3 Context Menu

The MagicLeap development team suggests keeping the context menu clean, and to not overload it with controls. It should only have relevant information and should be "a place to learn". They also suggest having it mapped to the bumper.

Figure 23: An example of direct targeting

3 Related Works

3.1 Related AR Works

In this section I will describe multiple AR applications which share features I aim to implement in the application. Here I will focus only on AR multiplayer applications which allows users to communicate and collaborate while manipulating 3D models in some way.

3.1.1 Spatiate

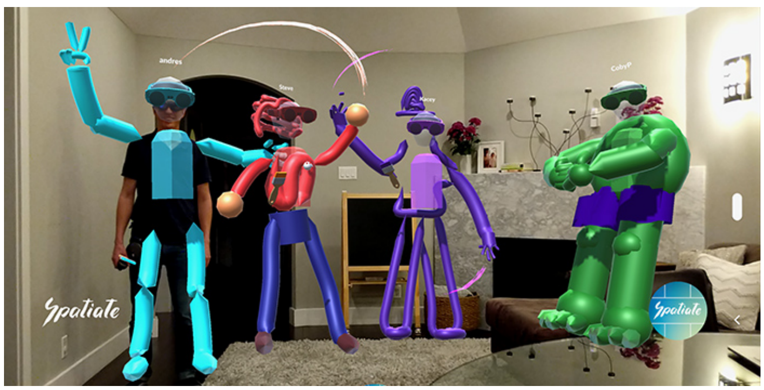

Figure 24: A collection of users posing inside sculptures they made. Their digital avatar is a 3D model of the magic leap HMD. The sculpture below is not part of their avatar.

Spatiate is an "art canvas" used to draw and create rough sculptures in 3D space. It is intended to be used with the magic leap, but there is also a mobile app in development. It was released in 2019 but is currently unavailable in Norway. This is an issue as it does not allow me to test the capabilities of the program any further than looking at the promotion videos available, as well as their website. Spatiate is included in this section because it is similar to the intended end product in that users can cooperatively build 3D scenes, add models, export, etc. Additionally, it is on the same HMD as I intend to use, and can serve as an example of how 3D avatars on the MagicLeap One can be tackled.

The Spatiate application has a heavy use of radial menus. It is used for choosing tools, colors, shapes, etc. These menus are opened using gestures with the non-dominate hand, while the dominant hand holding the controller can interact with the menu.

There is a built-in function which allows the user to export the model directly to the website SketchFab, which is used for viewing, selling, and downloading 3D models created by other users on the website.

Users can paint with different textures and shapes in 3D, as well as spawn different simple geometric shapes such as spheres, cones, cylinders, and cubes. These can be created with custom colors chosen by the user. The user can also paint with specific effects such as "glow" which makes the pain stroke emit light.

Spatiate allows for multiple users to work together, both colocated and remote. They are represented by a digital version of their headset, and a paint brush which represents their controller. Additionally, there is a cylindrical shape below their headset to indicate the position of their body. This allows for other users to feel their presence and allows for a higher degree of communication. It is unclear whether the headsets in the same room are spatially aligned or not, but it seems that way in the promotion videos. When displaying multiple users in the promotion videos, their digital representations are drifted slightly from their physical world user. This could either mean that the spatial alignment is not entirely correct, or it could mean that it was filmed with only one of the HMD's cameras, meaning it only looks off from that angle.

As far as the marketing material shows, there is no object transformation tool, other than when you initially place the item. There is also no gizmo, thus not allowing for very precise transformations.

3.1.2 Spatial

Spatial is an AR app designed to expand on the "Skype meeting" concept that can be used on the Hololens, Magic Leap or Oculus Quest. Like with Spatiate, there is also a mobile application. By giving each user a digital avatar to place in the room, it is intended to make them feel more "present" for the other users than they would if they were in a video call. Spatial is included because it includes an example of how to implement 3D avatars for users in an AR business setting. It also includes examples of how to tackle 3D model manipulations in AR.

Figure 25: An example of what a meeting looks like in Spatial. Featuring a 3D avatar for the person who is not there, and a virtual post-it board on the wall.

Multiple people can join one meeting, and each will be given a digital representation which consists of a torso with a connected hair and arms. The avatar is created from a photo of the user which therefore looks semi realistic, with an approximate resemblance to the user. The avatar follows their head and hand movements, making it seem like they are there in the room.

Using gestures, users can add post-it notes and move them around in the room. These can be categorized and used as a sort of kanban board. Users can also import and move images. Both the notes and images can be placed on a shared wall which acts as a "central hub" of information, meant for everyone who joins to see.

Images and 3D models need to be imported in order to be used in the application and have a limit of 30 Mb. Despite the size limitation, this means there is quite a bit of freedom in what can be brought into the meeting. Once imported, an object can be manipulated by performing a "grab" gesture and moving the hand around.

3.1.3 Ares

Ares is an MR application produced by HoloLight, built for MR HMDs such as the Hololens where multiple users can import and manipulate 3D models together. Ares is currently under development and new features are coming out regularly. The application as described in this thesis is based on information available in 2019. I am including Ares because it is intended for use in meetings and to collaboratively inspect and manipulate 3D models in AR.

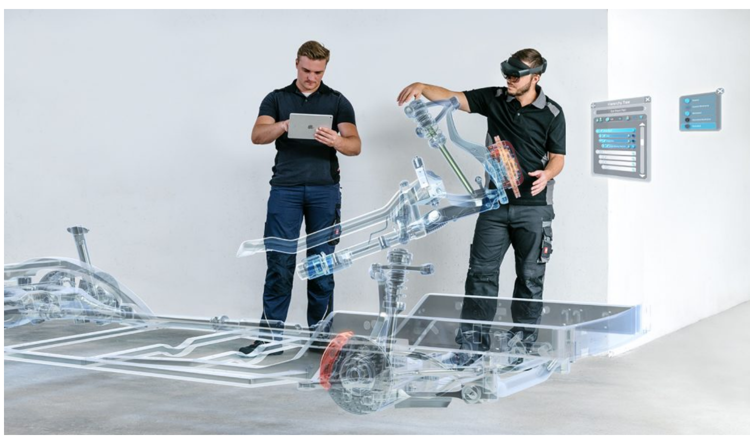

Figure 26: A visualization of what it looks like for a single user to manipulate a 3D model with Ares. A second user is looking at the manipulation through an iPad.

Ares does not feature any 3D representations of other users, but instead relies on the users being able to see each other in the physical world. This means the users must be colocated.

According to their teaser trailer, users will be able to see each other manipulate 3D objects in real time. Manipulation is done with hand gestures instead of controllers. The manipulations include moving, rotating, scaling, slicing and dicing models. [46] It is not clear whether the HMDs are spatially aligned or not, but one can assume this is the intended behavior based on figure 26

Multiple models can be imported simultaneously through the UI in the app. Simulations using CAD files can also be run in-app. Additionally, users can draw in 3D space, and save the scene with custom per-user settings [47].

3.2 Related VR Works

In this section I will describe applications which are not quite as related but can still work as examples on how to tackle user interfaces for manipulating and creating 3D objects. These applications are in VR but include elements such as 3D avatars for remote users, different 3D modeling and manipulation tools, and menu systems.

3.2.1 SculptrVR

SculptrVR is an application written by independent creator Nathan Rowe for use in VR. It allows for multiple users to enter a room together and create 3D models simultaneously.

Figure 27: Two users posing with a sculpture they made in SculptrVR

Users are represented by a floating 3D cylindrical torso which stays beneath the headset, 3D representations of their controllers, and a custom head model sculpted in the application by the user.

Users can see each other sculpting 3D geometry in real time. This includes adding and carving out geometry with different brushes, grabbing and moving geometry, copying and pasting, and painting geometry. There is also a voice chat feature which allows the users to communicate verbally. Together, this allows for users to collaborate on 3D art projects together.

Being VR instead of AR, the application does not require any kind of synchronization between physical world location and virtual location for multiple users. This means for a second person, there is no disconnect between the real person and the person they see in the application. This can be a good thing but it also makes the application less relevant as my aim is to explore how including the physical person in the application potentially elevates communication and collaboration.

3.2.2 Tilt Brush

Tilt Brush is a single player 3D painting VR application initially released in 2016 by Skillman & Hackett but now owned by Google. Using motion controls, the user can paint shapes in 3D space. It is effectively an approach to 3D modeling requiring less technical experience from the user. The menu is anchored to the non-dominant hand, as the dominant hand is used for painting. I am including Tilt Brush in this chapter because it is an artistic VR approach to 3D modeling and includes a creative take on 3D modeling user interfaces.

The menu itself is a cube which can be rotated, with a different section of the menu on each side of the cube. There is also a section below the cube for additional buttons. Tools in these menus include color pickers, importing and exporting models and drawings, and brush sets.

The painting itself in Tilt Brush originally consisted of only ribbon-like strokes which used the angle of the controller to decide the orientation of the ribbon, as well as the pressure on the trigger and the speed of the controller movement to decide the width. Later however, the ability to sculpt 3D models was added by allowing the user to add surface details to a primitive 3D shape by "painting" the new boundaries of the new surface, effectively painting volumetrically [48]. See figure 29 for an example.

There is a tool called "selection tool" in Tilt Brush where users can select objects with the trigger. After selecting, the object can be deleted with a "throw away" gesture or copied with one of the controller buttons. For more selection options, users can hold down the trigger to activate a pop-up menu which shows the options to invert the selection or select all. To move a selection, users can use the "grip controls" on the side of the controller to grab and move [49].

3.3 XR research projects

This section will describe multiple different ways researchers have tackled 3D navigation and manipulation in AR.

3.3.1 Immersive Group-to-Group Telepresence

Beck et al. (2013) [50] performed a study on group to group telepresence by having multiple users communicate with each other through virtually reconstructed life-size 3D representations of the remote users viewed on a screen with HMDs . They used multiple depth cameras to reconstruct the environment as well as the users in 3D. This experiment is included because it serves as an example of a less traditional approach to AR cooperation and 3D navigation using gestures and controllers.

Telepresence is defined as remote manipulation combined with high quality sensory feedback, later redefined as a term within telecommunication. Buxton et al. (1992) [51] argues that it needs to be divided into a person space and a task space. The quality of telepresence therefore depends on the quality of both. Beck et al. (2013) [50] argues that little work has been done since this initial research on bettering shared spaces for groups, and few applications include life size reconstructions of the user.

Figure 28: The Tilt Brush menu on the non-dominant hand. The menu facing the player is the currently active menu. It can be turned around to switch to one of the other menus which are visible behind it.

Figure 29: "Hull brush studies" by user Naam on Poly.google.com

3D navigation and manipulation was done with gestures and a shared input device called the Spheron which consisted of a trackball and a joystick.

Beck points out the importance of gesturing, specifically pointing, and the issues the team had with both the difficulties of aligning the physical world fingers with the 3D models they pointed at, and the issue of inaccuracies with the 3D reconstruction of the hand or fingers. Despite this, the users reported that communication through pointing assisted by voice chat communication worked well.

In the conclusion, Beck et al. (2013) [50] points out that although user feedback was positive, there were multiple misunderstandings in gestures due to inaccuracies in the 3D representation of users, and the dissonance between the virtual and real space. It was difficult to see exactly what was gestured towards, so they would have to guess and make estimations. The 3D representations were lacking, and thus could not compete with actual colocated users. In regard to the 3D representations, Beck argues that the subtle body language provided through their procedural reconstruction is more important than the reliability of pre-made, less dynamic 3D avatars.

The controls were simple in design and reportedly the users understood how to use the program "intuitively". I would however argue that this could be attributed to the simplicity of the navigation and manipulation system. Had there been more complicated 3D manipulation required, there is a possibility a more complicated control system would have to be implemented. As Beck mentioned, the gesture recognition was not reliable [50] and it can therefore be assumed an implementation of a Gizmo would be difficult to control with gestures.

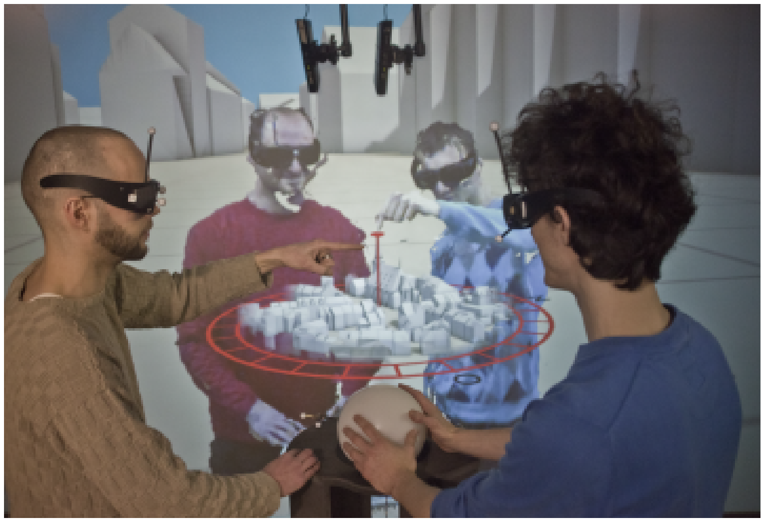

Figure 30: A group of users interacting with the 3D minimap for navigation. The remote group is projected on the wall

3.3.2 IEEE AR 3D art application contest

In 2017 IEEE held a contest in which multiple developer groups created AR tools for 3D art creation. In this section I will go through three notable examples in relation to AR 3D manipulation tools. They all use touch screen interfaces which is out of this scope, but they are included because they share various examples of 3D manipulation in AR.

Firstly, is the Batmen Beyond project. UI and AR objects are seen through the camera and screen on an iPad, anchored to a physical space tangible object. It uses a half pie menu to select different operations such as grab (comparable to the Grab Mode introduced in section 4.4.1), scale, copy and delete. According to the developers this is in order to avoid a complicated user interface as it would introduce cognitive overhead. Additionally, there is a 6DOF marker called "BatWand" which would serve as the cursor and would allow the user to manipulate objects in 3D. Users reported this as intuitive, however having to hold the iPad throughout put strain on the users arms.[52]

Secondly is SculptAR, which similarly to Batmen Beyond can place 3D objects anchored to a tangible object. The biggest difference being it only uses a mobile device and no additional marker like the BatWand. Instead, the "marker" is a cursor in the middle of the mobile device display. Operations are performed through touch screen gestures and the physical position of the device in combination with pushing different on-screen buttons. The main operations are painting, removing, copying, undoing and redoing, rotating, scaling and moving. [53]

Lastly is AACT: A Mobile Augmented Reality Application for Art Creation. Like SculptAR it relies on a touch screen UI on the mobile device and includes the following operations controlled with the position of the device and touch screen gestures: Translation, rotation, scaling, undo and redo.[54]

3.4 Feature Table

The following tables 3.4 and 3.4 are meant to compare the different existing works with each other and give an idea of what the intended end product is like. "Intended end product", the first element in each table, refers to the application developed in this thesis. As we can see, the plan is to implement what we deem to be the relevant parts from existing works: multiplayer, use of controllers, a 3D avatar. voice chat, 3D manipulation, 3D drawing, a Gizmo, and multiple operations.

Application	**VR**	**AR**	**Multiplayer**	**Controller**	**Gestures**
Intended End Product		x	x	x	
Spatiate		x	x	x	
Spatial		x	x		x
SculptrVR	x		x	x	
Tilt Brush	x			x	
Ares		x	x		x
Immersive Group-to-Group Telepresence		x	x	x	x
Batmen Beyond		x		x	
SculptAR		x			x
AACT		x			x

Table 1: Table of feature comparisons between applications

Application	**Spatial Alignment**	**3D Avatar**	**Voice Chat**	**3D Sculpting**	**3D manipulation**	**3D drawing**	**Gizmo**	**Operations**
Intended End Product	x	Head, torso, hand	x		x	x	x	Spawn object, delete, reset object, duplicate
Spatiate		Head	x		x	x		Spawn object, delete
Spatial		Torso, head, arms, hands	x		x			Spawn object, delete
SculptrVR		Torso, head, hands	x	x	x	x		Spawn object, delete
Tilt Brush			x	x	x	x		Spawn object, delete
Ares	x		x		x			Spawn object, delete
Immersive Group-to-Group Telepresence		Life Sized Reconstruction	x		x			
Batmen Beyond					x			Spawn object, delete, duplicate, undo, redo
SculptAR					x	x		Spawn object, delete, duplicate, undo, redo
AACT					x			Spawn object, undo, redo

Table 2: Table of feature comparisons between applications

4 Implementation and Design Choices

4.1 1st Iteration Requirements

This section explains the target audience as well as the product requirements. The requirements are a result of studying similar applications, and meetings with design students as well as multiple iterations of user testing.

4.1.1 Target Group

The intention behind the application is to make a generic product which can be developed further and tailored for different end users, preferably in an educational setting. As such, I was able to use a wide variety of visitors to the IMTEL lab as potential end users. These included architects, artists and city planners as described in section 1.1.1, but also NAV employees and developers, as the application could potentially be tailored to them as well.

4.1.2 Requirements

Before user testing, the initial requirements list was a very simple set of features as the initial purpose was to expand on the functional requirements through user tests.

Functional Requirements

- Multiplayer:
  In order to be used as a learning and teaching tool, the application needs to be used by multiple people at the same time. Both colocated and remote, but with a focus on colocated tasks.
- 3D manipulation of objects
  For technical users, there should be a Gizmo which seems familiar to them
  In order to perform quick and easy transformations without precision, users should be able to just pick up and grab objects.
- Spawn and delete 3D objects:
  In order to be able to build something new, the users need to be able to add shapes and objects to the scene, which they can then manipulate. The users need the option to delete these objects as well.

Non-Functional Requirements

- Someone who has never used the application before should be able to use it themselves:
  In order to save time and lower the barrier for users to use the application, it should be intuitive enough for new users to step into it without needing help from a second person.
- The Gizmo needs to be non-threatening for new users
- The application should be on MagicLeap One

- The operating system of the hardware must be continuously updated so that it is compatible with the latest development platforms

4.2 1st Iteration Development

In this section I will go through the different design and developmental choices made through the first iteration. Feedback on the first iteration can be seen in chapter 5.

4.2.1 Controller Scheme Challenges

Here I will go through the adjustments I made to the controller scheme suggestions made by the MagicLeap development team in section 2.9.1

Manipulation Controls

First is the manipulation controls suggested. As these are meant to be kept simple, I took them into consideration for the Grabmode, as that too is meant to be simple. The Gizmo is a more complex tool and requires more complex controls.

Their suggestions seem to be context based, or at least seem to assume that the user will not need to do all the transformations in one app. For example, several transformations require the use of different gestures on the touch pad. Through testing I quickly discovered that it was difficult for users to perform a gesture without the Magic Leap confusing it for another. I also discovered that using the touchpad is one of the first things a user will try. It makes sense to reward this behavior by adding more functionality to the touchpad. The challenge then becomes, how does this need to be changed in order for the user to be able to do all these transformations without the overuse of gesture? For this reason, I decided not to apply more than one functionality to the touchpad at once. This means that whenever a user has a tool active, the touchpad will only serve one function. This removes the possibility of users mistakenly performing one gesture over another.

In grab mode, the user is able to rotate the object by grabbing it and rotating their hand. It is therefore important to consider whether a rotation functionality with the touchpad in grab mode is necessary, since it is already available.

The MagicLeap development team suggested adding scaling by tapping the left and right sides of the touchpad, but I left it out in order to keep the simplicity of the touchpad.

I decided to go with the MagicLeap development team's suggestion of adding "push and pull" to the touchpad in grab mode. Meaning users could touch the upper part of the touchpad to push an object away, and the lower part to pull it in. The amount of pushing and pulling was decided by the distance from the center.

The "Grab and Place" behavior described by MagicLeap is consistent with how I planned to implement the Grabmode. This is also consistent with how the Gizmo Tool works, except there is another layer to it. The Gizmo Tool is grabbed and held with the trigger, thus affecting the highlighted object.

Content Selection

Because the functionalities in the application require the use of multiple buttons, this is where some more creative liberties had to be taken. As the trigger needed to be reserved for activating transformation tools, it would not be used for selection when selecting 3D objects. This is a result of user testing where users would become confused and frustrated if the trigger could be used both for transforming and selecting. For this reason, the bumper would be used for selection instead of the trigger. Like suggested, the users became frustrated if they had to switch to the touchpad when navigating menus, thus the menu would be navigated with both the touchpad and the ray. In addition to the ray, users could also select objects by putting their controller into it and tapping the bumper.

Context Menu

Magic Leap suggests mapping the context menu to the bumper. However, since the bumper is mapped to selection, the context menu will have to be mapped to something else. The only other available button is the Home Button, which Magic Leap suggests being used as a "back button" and as the application home menu. The context menu will therefore contain a section which works as a home menu, in addition to the context sensitive part.

4.2.2 Implemented Features

The first prototype consisted of a selection of models and a controller with two modes in which the player could move around, scale, and rotate the models in the environment. This was done as a way to get used to the software and hardware, as well as being used as a proof of concept when presenting and gauging interest in the project. This prototype would only be available for one person at a time, thus only being able to test how intuitive the controls are.

The UI consisted of a ball connected to the origin point of the controller, which would act as the cursor. A piece of text next to the ball would signal which control mode was active when visible. When the Gizmo was active it would be invisible, and while the Grabmode was active it would display "Grab mode on", see figure 31. This was mainly done for debugging and for users to have a visible distinction between interaction modes as there was none at the time. Pushing the home button on the controller would open a menu which would allow the user to exit or switch control mode.

The two control modes consisted of one using the Gizmo, and one using a more intuitive tool-free approach. These would be called "Gizmode" and "Grabmode", and were what the user would use to transform the selected 3D object.

Selection was mapped to the bumper. When pushing the bumper, it would check if the controller was intersecting with an object, in which case it would select it. If holding down the bumper, it would cast a ray and check which objects the ray was intersecting with. It would select the object which most recently intersected the ray. This was to compensate for the fact that objects would often be far away from the user.

Figure 31: A screenshot of a user selecting an object with the tap function, while Grabmode is active.

Gizmode

Gizmode was intended for more precise operations, such as moving the model slightly on one axis, or rotating it slightly on one axis, etc. Meanwhile, Grabmode was intended to be more straight forward and intended for users inexperienced with 3D graphics programs.

There were several variations of Gizmo behaviors in this prototype, as the behaviors of a gizmo on a 2D screen would not necessarily translate into 3D. When selecting the arrows of a gizmo on a 2D screen, the user does not have to be "close" to the gizmo as the distance becomes irrelevant in 2D. However, in 3D, the user would have to account for the distance between the 3D position of the controller, and the 3D position of the gizmo. These two designs will be referred to as Gizmo1 and Gizmo2.

The main difference between the two gizmo designs is their movement in relation to the controller, as well as some interactions in different transform modes. Commonalities between the two are the visuals, where they both used the same 3D axis models, and the control scheme. In both modes at this point in development users could use the touchpad would cycle transform modes i.e. scale, rotate, translate.

Gizmo1 (See figure 32) was similar to the "3D Cursor" option as a pivot point in Blender 2.8, see section 2.7.3. Gizmo1 was essentially not a separate control mode, but rather an object which could affect other objects in the scene through the user interacting with it using the Grab Tool. When starting the application, Gizmo1 would be at the center of the application i.e. location (0,0,0). To transform an object with Gizmo1, an object would have to be selected, and the user would have to grab an axis or the center point with the trigger to move the object the same amount the gizmo is

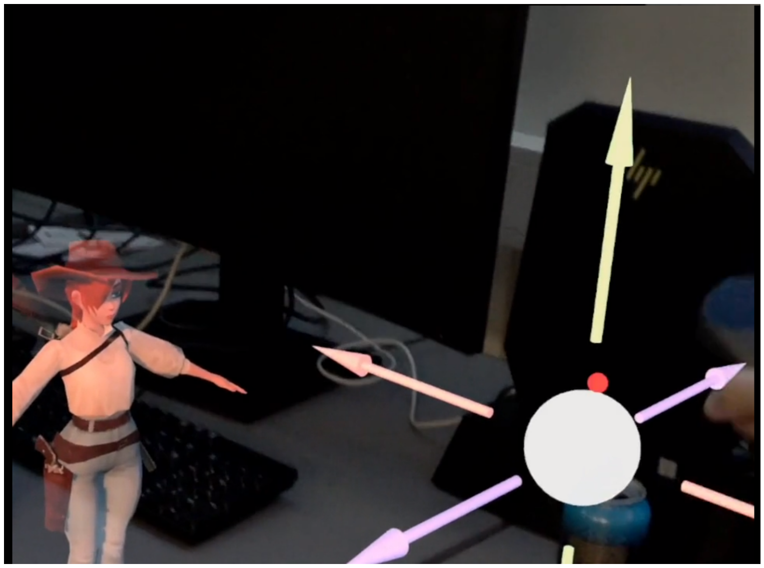

Figure 32: A screenshot of Gizmo1. At this point in development, the Gizmo acted independently of the controller. The red dot indicates the controller location. Depicted here is a user grabbing an axis on the gizmo to lock movement on said axis.

moved. When letting go of the trigger, Gizmo1 would snap back to the original position. This was done in order to prevent users from having to physically walk around with the Gizmo when moving an object far away. In order to move Gizmo1 without it snapping back, the user could hold down the bumper for it to snap to the controller. When letting go of the bumper, Gizmo1 would stay in place. This also meant that activating the selection ray was not possible as long as Gizmo1 was active. To deactivate Gizmo1 and enable the selection ray, the user would push the home button. To lock movement on an axis, the user would simply intersect the axis with the cursor and grab it with the trigger. To transform on all axes, the user would grab the center point of Gizmo1. This design was the most similar to traditional 3D editing software as described in 2.8.2. In order to switch transform modes, users would tap the touchpad to cycle between them.

Translation with Gizmo1 was done by dragging an axis. Gizmo1 would follow the position of the controller, but only on said axis. Rotation was done by grabbing an axis and "rotating" it. Gizmo1 would track the position of the controller and spin towards it as if the user was grabbing and rotating a wheel. Scaling was done by grabbing an axis and stretching it. Gizmo1 would stretch said axis towards the position of the controller.

Gizmo2 (see figure 33) would follow the controller, hovering about 5 cm above it and acting as a separate control mode from the Grab Tool. This meant that the user did not have to hold down a button for it to snap to the controller as it would always be at the controller. By extension, this means the user did not have to deactivate Gizmo2 in order to use the selection ray. However, it also means the user would not be able to lock transformation on an axis the same way. Selecting

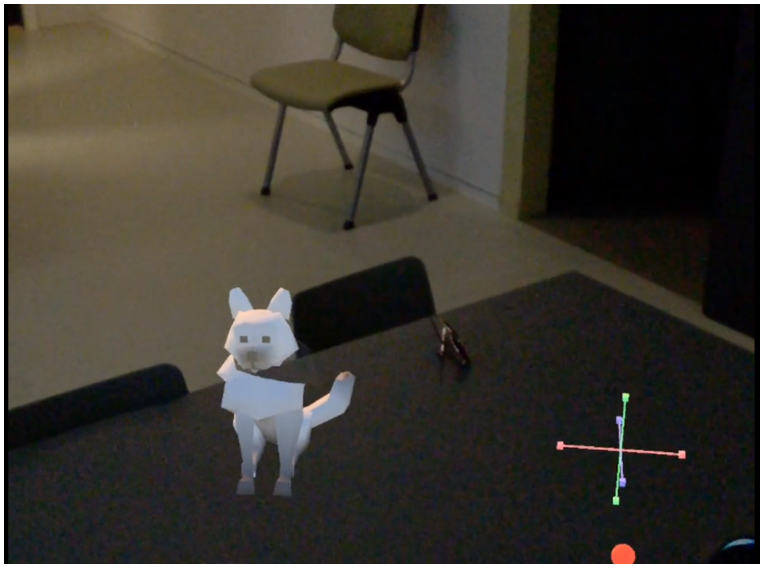

Figure 33: A screenshot of Gizmo2. Here the user is scaling an object on all axes. The location of the controller is indicated by the red dot.

axes was instead done by tapping on the left side of the touchpad to cycle through them. Changing transformation modes was done by tapping the right side of the touchpad. Swapping between Grabmode and Gizmo2 would be done by pushing the home button.

Transformation with Gizmo2 would happen on the selected object regardless of where the controller was when the trigger was pushed down, as the user did not have grab axes anymore. The gizmo would copy translation on a given axis similarly to with Gizmo1. Scaling would also be calculated similarly to Gizmo1. Two different attempts were made for rotation, they will be referred to as rotation1 and rotation2. Rotation1 would track the location of the controller and aim the selected axis of the gizmo toward its location. Rotation2 would copy the rotation of the controller. See 38 for examples.

Grabmode

In Grabmode, the selected object would become a child of the controller while the trigger was held down. When letting go of the trigger, the object would stay in place. This is to imitate grabbing the object with your hand. Due to the object inheriting rotation and position from the controller it would move at high speeds around the user but moving it away or closer would be slower. To compensate for this, the touchpad could be used to pull the object closer or push it further away at higher speeds than what the user could do by moving the controller in and out. Due to time constraints, there was no implementation of scaling in Grabmode.

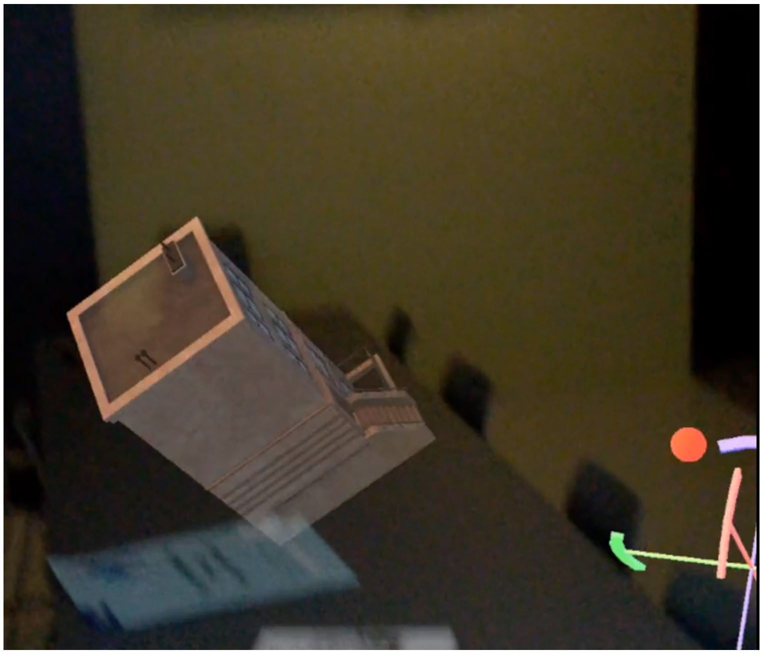

Figure 34: A user rotating a building on all axes using Gizmo2 in combination with rotation1

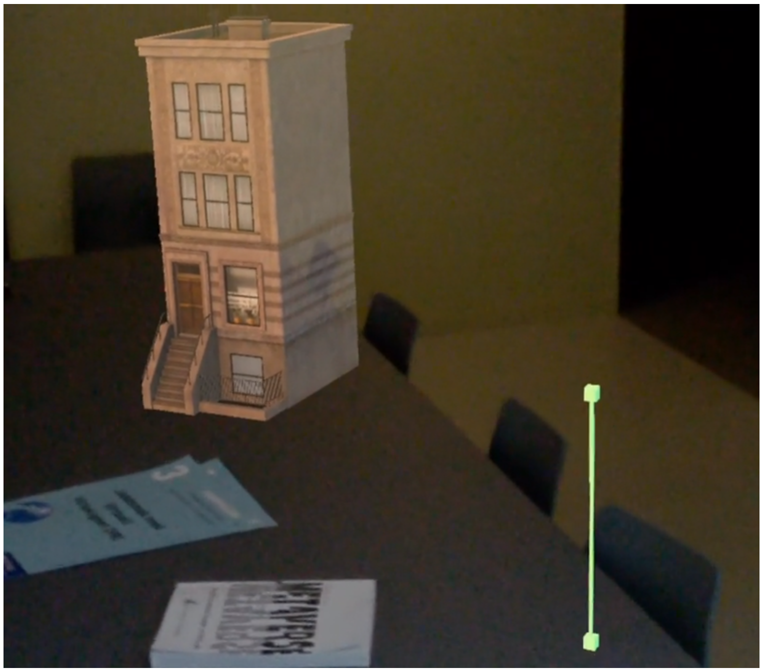

Figure 35: A user scaling a building on the Y axis using Gizmo2

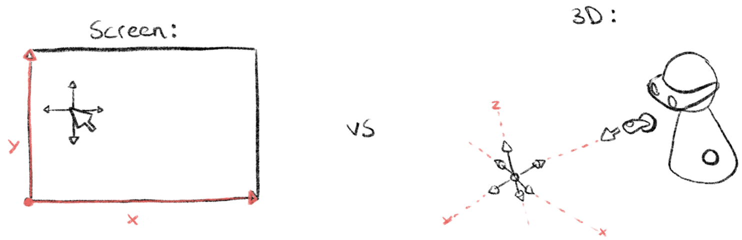

Figure 36: An illustration of the problem of selection in VR versus on a screen. The third axis also creates a third distance which has to be taken into account during selection.

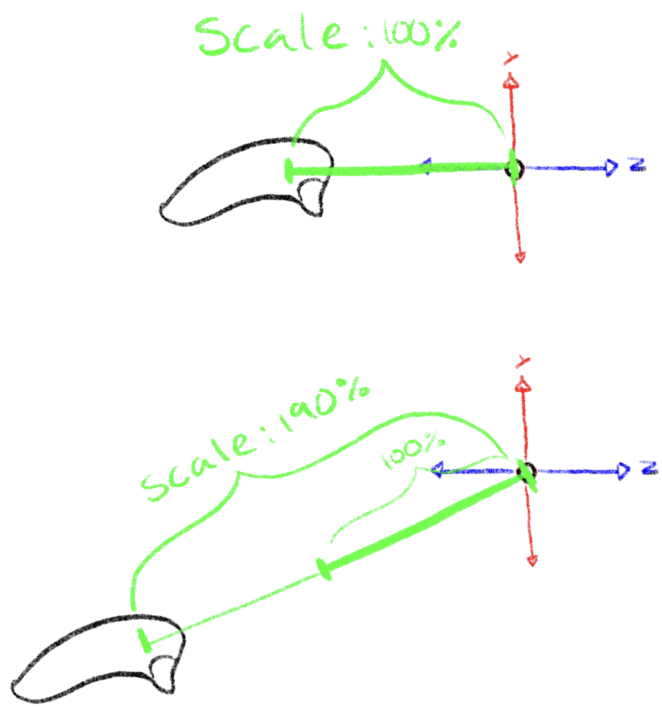

Figure 37: The first implementation of scaling with the gizmo. Pushing the trigger calculates the distance to use as 100% scale. As illustrated, moving the controller will then calculate the difference and use it as a scale factor.

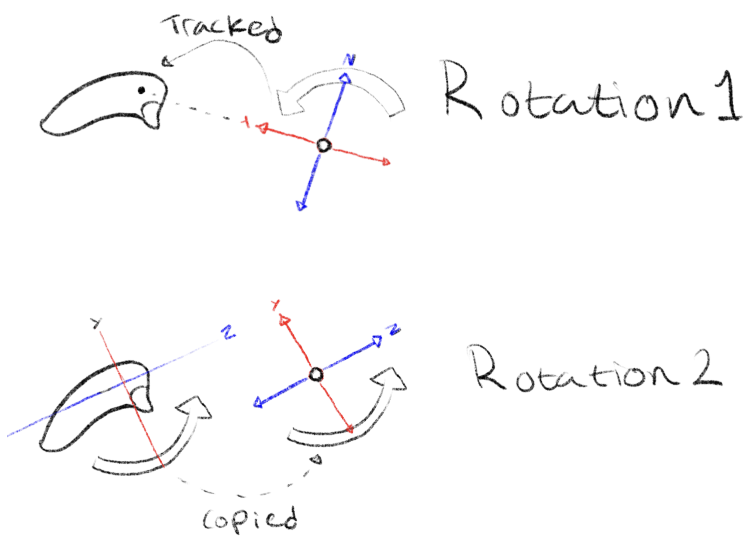

Figure 38: An illustration of Rotation1 and Rotation2. Rotation1 aims towards the controller, while Rotation2 copies the rotation of the controller.

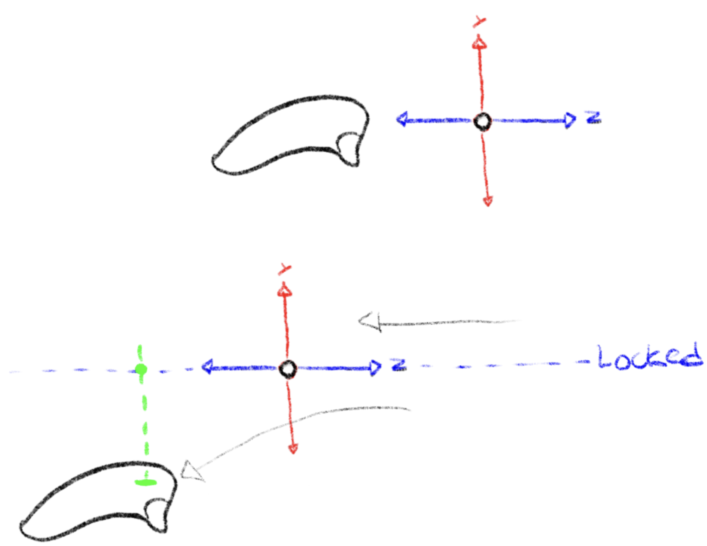

Figure 39: An illustration of how translation worked with the gizmo. The gizmo would copy translation from the controller, locked on the selected axis.

4.3 2nd Iteration Requirements

For the second prototype, a new, bigger list of requirements was put together keeping in mind feedback from the first prototype in section 5.1. I have added explanations for the requirements which are new.

4.3.1 Functional Requirements

1. The UI needs to mitigate the impact of the limited field of view and the unfamiliarity of the MagicLeap One controller:
   Many users during user testing commented on the field of view of the magic leap. The MagicLeap One has a relatively small field of view as will be further discussed in section 2.4.2. This means I need to make the most of the visibility there is.
   The Magic Leap controller is a new and unfamiliar controller to the average user as seen in in section 5.1.2. The buttons are without labels, and during user testing I quickly discovered that most users do not know the difference between a bumper and a trigger. The buttons are also not labeled, which makes it more difficult for the user. After showing the location of every button, the user still quickly forgets the location, or the fact that the button existed at all. For this reason, the UI needs to show clearly where each button on the controller is, and what each button does.
2. Manipulate 3D objects:
   Users need to be able to grab a 3D object and move it around, as well as scale. This needs to be done in a quick way for casual users as well as a precise way for technical users. It was important that the Gizmo was not the default interaction method, as I noted in section 5.1.2 some inexperienced users were confused by the Gizmo.
3. Multiplayer
4. Add and delete 3D objects
5. Select objects from up close and afar:
   In order to speed up the selection process, the user needs to be able to select an object from afar instead of having to go up to it. The user also needs to be able to simply reach out to an object that is within reach to select it. This was added to the requirements under development of the first iteration because of the immediate problem of not always being close to objects. This was added because it was not explicitly in the requirements before.
6. Gesture and point at objects:
   Gestures are an important part of human communication, and by extension teaching. A user therefore needs to be able to use their hands as a means to communicate and appear more present to the other users [11]. This was not in the previous requirements, but after researching it was clearly an important feature.
7. Hide and un-hide objects:
   While making intricate objects, some parts can often get in the way. That is why it is often useful to be able to hide an object to see another, or to work on another object without having to worry about the hidden object obstructing. After hiding an object, it is of course important

to have the option to un-hide it. This was added due to suggestions during user tests.

8. Select objects from a list:
   When a project eventually becomes large and complicated, with many objects, some objects might get lost or become difficult to select. For this reason, it might be important to be able to select it from a list. This is also important in order to select an item that is hidden.

4.3.2 Non-Functional Requirements

1. Operations need to be simple to perform:
   Operations like selection, manipulation, delete, reset, duplicate, etc. need to minimize the number of buttons pushed and menus interacted with in order to perform. This should make the application less tiring by reducing the mental and physical work load.
2. The UI should avoid relying on memory:
   XA was confused by Gizmo1 (see section 5.1.1), because the controller scheme changed based on which tool was active. It was not immediately visible that Gizmo1 was active and if the Gizmo was not in her view, the application relied on her remembering it was active. The application should avoid situations like these as much as possible.
3. Someone who has never used the application before should be able to use it themselves:
4. Holograms must stay in place with minimal drift unless moved by the user:
   In order to maintain precision and a sense of immersion the holograms need to be steady and not drift away.
5. The application should run on the MagicLeap One
6. The operating system of the hardware must be continuously updated so that it is compatible with the latest development platforms
7. The Gizmo needs to be nonthreatening to new users:
   Not everyone is used to the Gizmo as a general concept unless they have experience with 3D manipulation from before. Therefore, the Gizmo needs to feel intuitive and lower the barrier of use

4.4 Second Iteration Development

In this section I will detail the changes made to the application as well as changes which I was not able to implement. The second iteration added multiple features as well as network capabilities. It also made changes to the UI. The developers of Magic Leap also released multiple tools which could possibly aid the development. They developed a tool for grabbing and transforming objects which could replace Grabmode and add scaling to it. 4.4.1 They also made tools for synchronizing the AR space with the physical space, see 4.4.4

4.4.1 Implemented Features

Due to user feedback from the first prototype, the second prototype was built with a combination of Gizmo2 and Rotation2. Selection was made clearer and more intuitive, a context menu was added, as well as a tool for drawing. Additionally, this prototype had multiplayer implemented so collaboration could be tested.

Resource Shelf

Instead of a 2D menu, a 3D shelf-like structure was built which would spawn objects inside cubes i.e. "shelves". Objects inside their respective shelf will rotate to signify that they are being displayed. Once dragging an object out from its shelf, the shelf will spawn a duplicate of the object so that it keeps re-stocking.

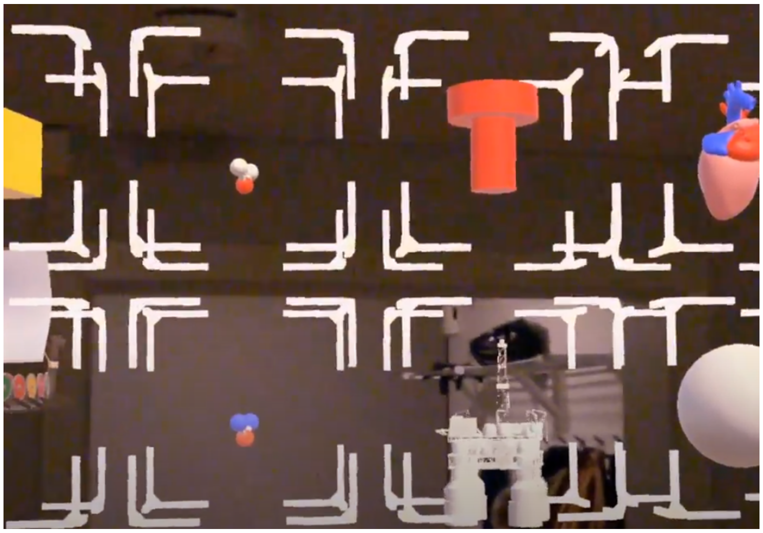

Figure 40: A screenshot of the Resource Shelf. Each box contains its own 3D model which it spawns

Gizmo Tool

Like mentioned earlier, Gizmo2 with Rotation2 was chosen for further development, and remained in large parts the same. The biggest differences were in the UI as well as optimizations and bug fixes. The new Gizmo UI used a radial menu fixed to the touchpad. With the radial menu, users could choose X, Y, Z or all axes, as well as change between Move, Rotate, and Scale mode. Touching the touchpad would enlarge whichever part the user held their thumb over, as well as update the Gizmo.

Grab Tool

The Grab tool was changed slightly from the previous iteration in that it now featured all transform operations. Instead of using the touchpad to move the object closer or further away, the touchpad was used to scale the object. This was implemented because most users reported scaling in grab mode not working as expected, despite there not being a scaling operation in Grab mode. This is likely because they were confusing an object moving closer or further away with it being scaled up or down.

Additionally, the grab tool itself changed appearance. The controller location was no longer

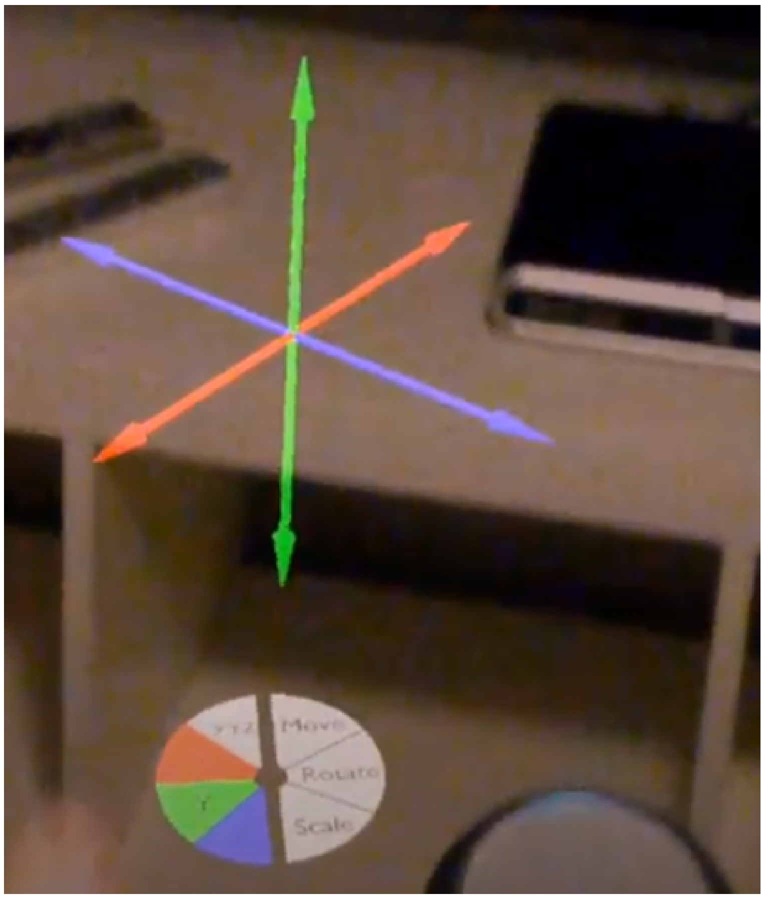

Figure 41: A screenshot of the Gizmo Tool in the second iteration.

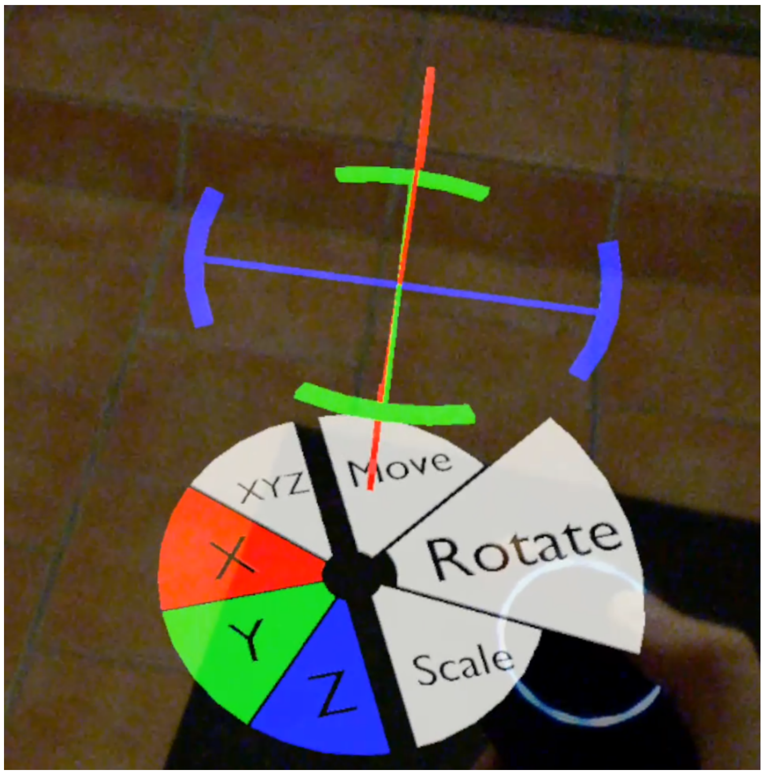

Figure 42: Here we see the user selecting Rotate on the touchpad while using the Gizmo Tool. "Rotate" is enlarged, which activates the rotation version of the Gizmo as indicated by the arcs on the axes.

signified by a red dot, but a ball with cone instead to display location as well as orientation.

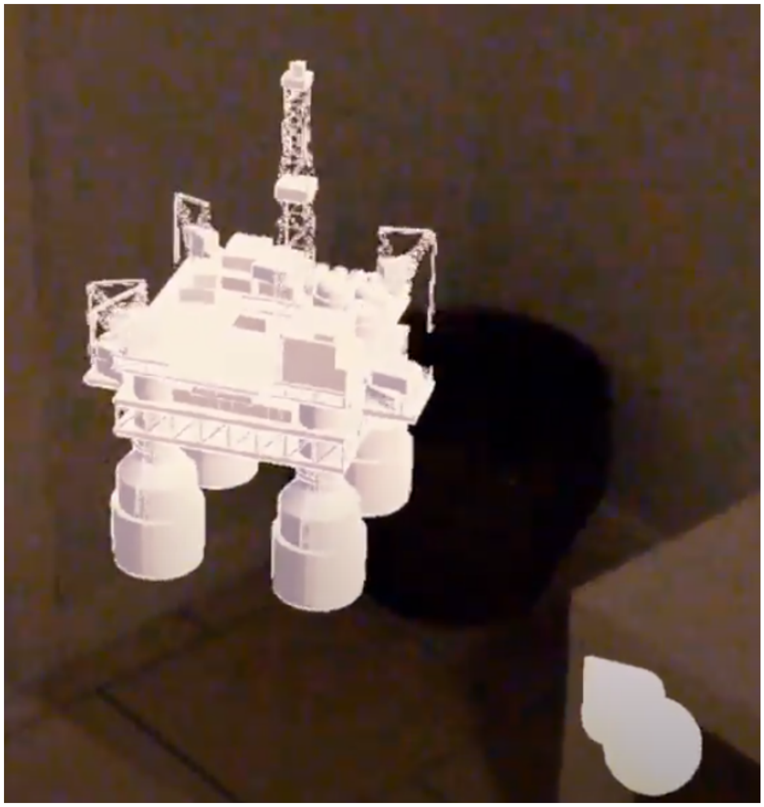

Figure 43: A screenshot of the Grab Tool. The white ball and cone indicate the controller is being pointed forwards

4.4.2 Multiplayer

The Photon libraries handle issues like server hosting and synchronization of local data across multiple users. This allows for a synchronization of the transforms of objects on all devices in the same server. When adding new objects, Photon relies on prefabs, which are predefined Unity objects, in order to add them dynamically to a scene across devices. An object cannot be added for multiple users without it being a prefab. Synchronization happens through Photon Transform Views which copy the transforms from the "owner" to the other users [55].

4.4.3 User Interface Changes

Some changes were made to the user interface to increase affordance and visibility. Most significant is the context menu, as well as the method to display selection.

In the first iteration, the Home button would simply swap the active tool. In the second iteration the Home button would open a context menu which allowed the user to swap tool, delete object, reset object, or quit the application.

The "delete" and "reset" buttons would update with the name of the selected object in order to make it clear which object would be affected. The context menu could be navigated by tapping the upper or lower part of the touchpad, to go upwards or downwards. To increase the affordance of the touchpad, arrows pointing up and down were fixed to it to signify that pushing them could navigate the menu.

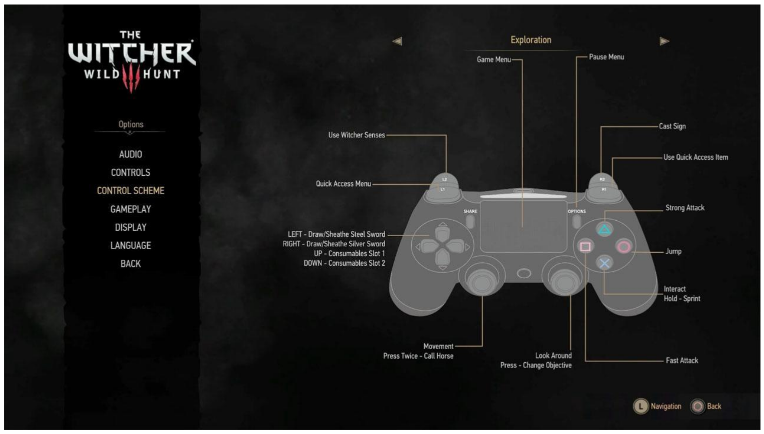

Figure 44: The controller layout screen from The Witcher 3 on Playstation 4

An option was added in the context menu to show or hide the button layout. The button layout was a 3D overlay over the controller. The overlay consisted of the names of each part of the controller, i.e. "trigger", "bumper", etc. Each name had an arrow which pointed to the respective part of the controller. The intention of this was to make it easier to familiarize the users with the controller, in the same way controller layouts are often visualized in video games. See figure 44 for an example.

In the previous prototype, selection was displayed through text in the UI fixed to the bottom left corner of the screen saying "selection:" followed by the name of the object. However, it was difficult for users to relate the text in the corner to what they were seeing in the application, as well as it not reflecting the industry standard for selection. To mimic other software, an outline was added to selected objects by applying a different shader to the model once the selection has been made. This selection shader would extrude geometry from the original mesh and flip the normals. Polygons in Unity are one-sided by default, meaning that viewing a polygon from the inside will make it invisible. This makes it so that the extruded mesh appears to be "behind" or "around" the original mesh. By changing the color of the original mesh to white, this effectively makes it appear as if the object has an outline. When deselecting or selecting a different object, the selected object reverts its shader back to the original.

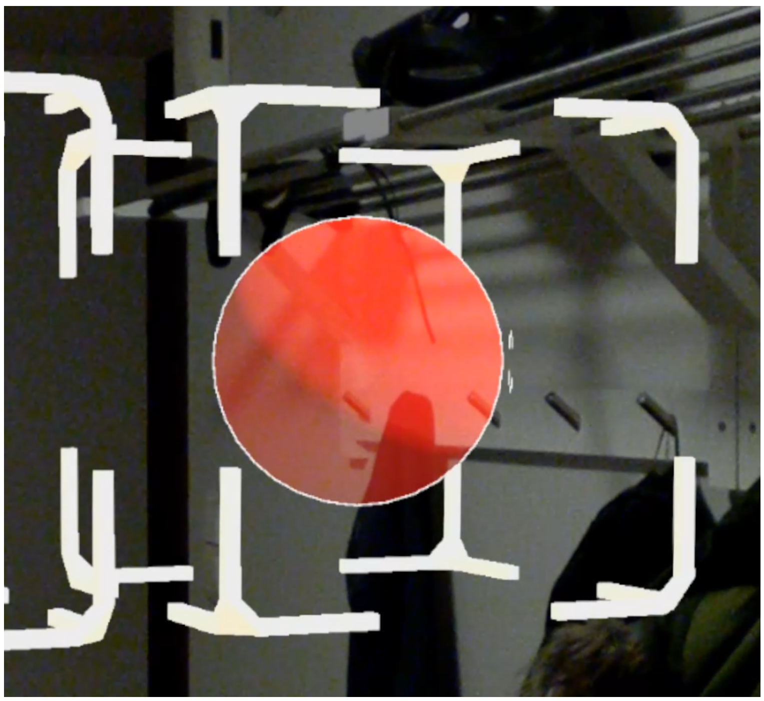

Figure 45: A selected red sphere. Selection is indicated by the white highlight around it.

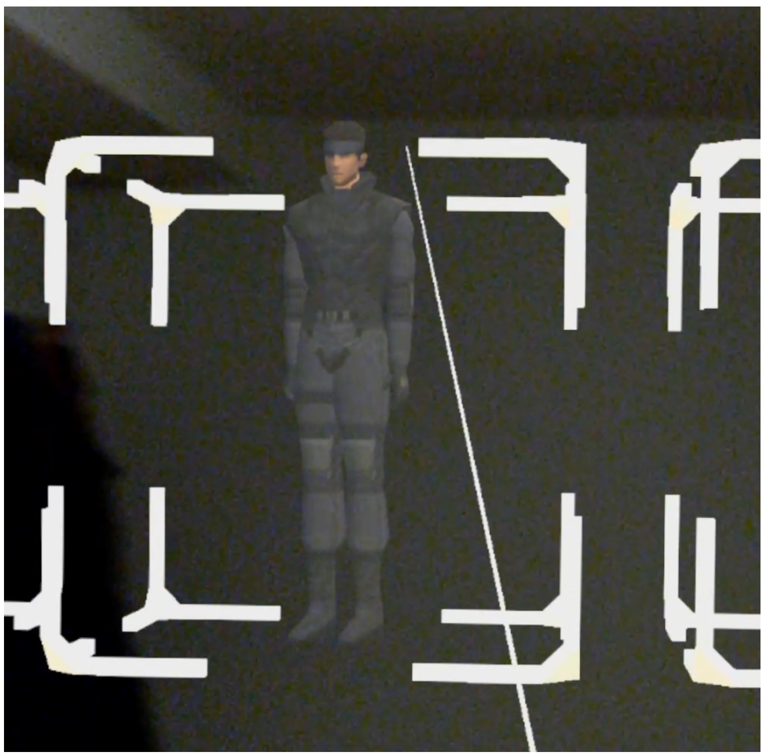

Figure 46: Here the user is aiming a selection ray towards a 3D model in the Resource Shelf

4.4.4 Dropped Features

In this section I will detail features which were on the requirements list, and intended for this iteration, but not implemented. Some were dropped because of technical difficulties, while some were dropped because of poor results in testing. I will go through each and explain why.

MagicLeap Toolkit

As mentioned earlier, there were new tools published by the MagicLeap development team. Initially these were supposed to be implemented and replace the Grab Tool. However, these were released simultaneously as MagicLeap adopted a new image and changed their MagicLeap package manager. Additionally, there were links missing in the official guides for implementing the tools which were solved after some back and forth with the developers. This all summed up in delays which ended up with other features being prioritized over implementing the new MLDK tools.

Sophisticated Multiplayer Event Handling

In this application ownership was programmed to be transferred when a user selects an object. In other words, selecting an object makes you the owner, which then means every other user's local copy of that object will copy the transforms from you. Ideally, there would be more logic to it, preventing multiple users from selecting the same object for example. However, the application still works without it, so like many other features this was deprioritized.

Spatial Alignment

Spatial alignment as the MagcLeap development team defines it, is when digital objects align correctly with their physical counterparts across multiple devices. Each Magic Leap HMD will have its own definition of where the origin point of the digital world is, as well as the orientation. This means that for two or more devices, they need to synchronize their origin points and orientation. The MagicLeap ToolKit (MLTK) Spatial Alignment does this by synchronizing with other devices on the same local network [56].

This in theory, is great for colocated work, and I initially assumed this would be simple to implement. If it were implemented, it would increase the workspace awareness as every user in the same room would agree on where objects are in physical space. Their 3D avatars would also align with their physical body. however, at this point in development the application was intended for both colocated and remote work. This would require more sophisticated network handling than what was currently implemented: checking each user for whether they are on the same network and aligning their AR world differently than other players. For this reason, MLTK Spatial Alignment was not implemented.

4.4.5 Alternative to Spatial Alignment

In theory, image tracking can be used to define a 3D location and orientation in physical space, which could then be used as an anchor for the AR world. This would mean any MagicLeap One seeing the same image on their table would have their anchor in the same position. However, Photon Transform Views which were used to synchronize object transforms across separate networks, synchronize global transforms by default. A more ideal solution would be to change the default code to use local transforms. That way the objects would not be synchronized for their absolute position within the AR world, but for their local position in relation to the tracked anchor. Development was started on an iteration of the software which was anchored to an image tracker.

This was however something which would take time to develop and test and was pushed down in priority due to Covid-19 reducing the importance of colocated collaboration. The spatial alignment would only be an issue with colocated collaboration as users cannot tell if for example a physical user is not aligned with their avatar when they cannot see the physical user. It was still possible that one colocated group would work with a remote group, and for the colocated group this would be an issue, but it was still scrapped to save time.

Color Wheel

Originally, there were plans to add a color wheel which would allow for a user to select a color to draw with, as well as an option to change the size of the brush thickness. Finally, there were plans to make the drawings selectable objects like the rest of the objects in the application. This would allow for users to move drawings around, and delete them or duplicate them, scale them, etc. These features were however cut to save time. Instead, the drawing tool would have a fixed brush size with a red color, which could not be removed after being drawn.

Voice Chat

Voice chat was initially not a very high priority feature as the focus was on colocated collaboration. After the Covid-19 outbreak, remote collaboration became more important and so voice chat became an important feature. Being able to hear each other and talk freely is a clearly important part of collaboration and communication, so having it implemented in the application would be a very important feature. Originally, the Photon Voice library was intended to be implemented. After studying the library and performing some simple tests, it became clear that the standard Photon Voice library would not work with the MagicLeap One. According to the Photon docs, in order to have it work on the MagicLeap One, Photon Voice needs to be expanded with an industry-circle exclusive add-on. Emailing the Photon development team about this caused more delays and complicated the feature too much, and lead to it being scrapped.

4.4.6 Tutorial Video

It is IMTEL tradition to add a tutorial to the project, it was also very clear during testing that the ML controller was too unfamiliar for most users, as well as the functionality of the application being too foreign. Ideally the tutorial would be an interactive part of the application, however to save time a video tutorial was made instead, using Blender 3D, and official Magic Leap 3D models.

One of the biggest problems during testing was to get a user to quickly understand where each button on the controller was, as seen in section 5.1. To combat this, the tutorial video had a heavy focus on the MagicLeap One controller.

The first part of the video was a turnaround of the MagicLeap One controller which pointed to different buttons, explaining their name and purpose. During this part, the users would usually turn their own controller around and take a mental note of the buttons.

The rest of the tutorial went through every transformation tool, explaining how it was used and which buttons to push. The controller in the video performed the transformations on a 3D object while a magnified version of the buttons was displayed on the side. This included selecting the model each time as well, in order to hammer down that an object needs to be selected in order to perform transformations on it. In order to make it more visible which button to push, each button that was pushed down was highlighted in green in the video. This coincides with the UI color scheme.

Due to the Covid-19 outbreak, the campus was locked down, meaning access to the video was lost for several weeks. This would end with the tutorial video not being used in the final user tests.

4.5 Adjustments Before Final User Tests

Some changes were made to the application due to feedback from user tests during the second iteration. These changes were made in preparation of the final user tests in order remove immediately visible issues which could worsen the results.

4.5.1 Context Menu

The third and final design of the context menu divided it into three parts, the first two being next to each other separated by frames, the third being appended below them. The left side was dedicated to the tool choice. Instead of "Gizmode" and "Grabmode", the tools were now simply called "Gizmo Tool", "Grab Tool", and "Draw Tool".

To increase affordance and visibility the name of the currently active tool was displayed at the top of the tool section of the menu. On the right side was the context sensitive buttons. Mirroring the left, the name of the currently selected object was on top, with buttons to delete, duplicate, and reset the selected object below. Unlike the previous iteration the text on the buttons would stay the same, only the name above them would change.

The third group of buttons were system related, and only consisted of one button which was the exit application button.

This menu could be navigated both horizontally and vertically with the touchpad, as well as by touching buttons with the cursor or selecting them with the selection ray. This was for selection methods to be more universal.

Draw Tool

In order to allow a form of annotation as an expansion on gesture, a drawing tool was added to the application. This was implemented by following a tutorial from the FusedVR YouTube channel [57] with slight modifications to make it work in multiplayer. It functions similarly to Tilt Brush 3.2.2 in that it creates 3D geometry based on the controller location. The angle of the 3D planes being drawn is decided by the orientation of the controller as well as the direction the controller is moving. A slight difference from Tilt Brush is that it uses controller velocity as a factor as well. The velocity of the controller decides whether to draw a line while holding down the trigger.

As mentioned in 2.6, objects need to be prefabs in order to be added dynamically across devices. 3D drawings, being procedural, could not be spawned by way of prefabs. In order for drawing to

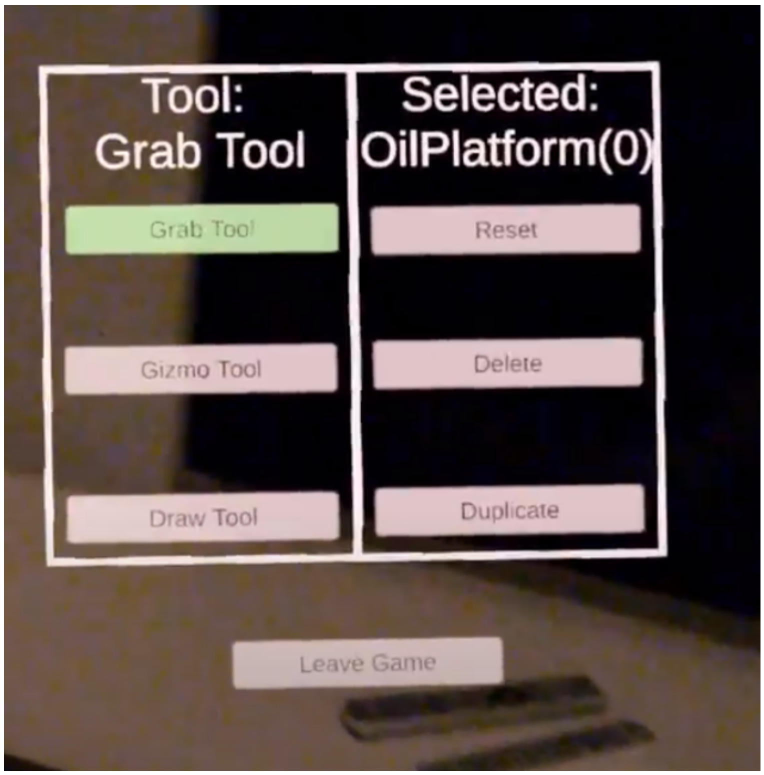

Figure 47: The final context menu

Figure 48: The Draw Tool in use.

work in multiplayer, every user has their own local drawing object, which calls the same functions for each user whenever a user draws a line. The Photon Transform View ID of the currently drawing user was passed to each user in order to draw in the correct 3D location. This way, sending entire 3D models over the network could be avoided.

Duplicate Operation

A third operation in addition to delete and reset was added to the context menu. The "Duplicate" button finds the prefab of the selected object and creates an identical copy in the same scale and rotation as the original, but a little to the side.

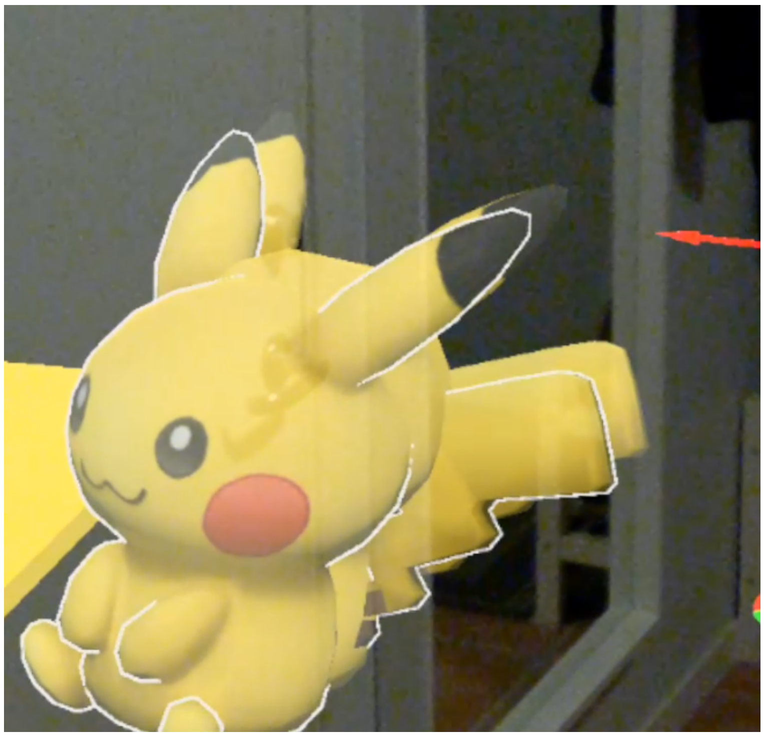

Figure 49: A duplicated pikachu model

5 Test Results Under Development

5.1 1st Iteration User Tests

The first prototype was tested by fellow students in the lab, visitors from different fields, and a couple field experts. The test consisted of two different stages. The first prototype mostly lacked a UI and would rely on either experimentation or narration of the controller scheme in order for the user to understand how to use the program. Therefore, the first stage consisted of instructions on how to use the Magic Leap, what the names of the buttons were, an explanation of what a gizmo is, and an explanation of the controls. In the second stage, the user would be placed in the application, surrounded by different 3D models. This included a building, and a dog. The dog was the same size as the building, and so the task given to the user was to scale and move the models so that the dog could fit through the entrance of the building.

5.1.1 Expert feedback

There were two experts willing to test the first iteration. The expert visits were less spontaneous than the student visits. Thanks to this, they had time to have conversation after the experiment and give additional feedback afterwards. While performing the experiment, they were instructed to think out loud so that I could take notes. I asked them about what they thought about the application in general, the Gizmo, the Grabmode, what they would change about it, and otherwise let them talk freely.

Expert evaluation by Art Project Director (XA)

XA had experience with directing and budgeting several art and science projects. She also had experience with the MagicLeap One as she had contacts within the MagicLeap company. She tested a variation of the application with Gizmo1 and Rotation1.

XA thought the Grabmode was "fine", and expressed it was easy to use, being able to move the dog and the house around. After activating the Gizmode, performing a translation, rotation and scaling operation with Gizmo1 she said this also was "fine" and "pretty basic".

After attempting to move the dog into the house however, she quickly became frustrated as she expressed, she expected the Grabmode behavior to still be active while in Gizmode. She was pushing all the buttons on the controller and asking why nothing was happening. Attempting to assist her, I asked if she could see the Gizmo or not, in order to determine if she was still in Gizmode, to which she frustratingly cursed at us and took off the HMD, ending the experiment.

It is fair to say that XA thought the UI for the first prototype was insufficient. She explained that there was not enough indication of what was happening, and why nothing was happening when she pushed buttons. She also did not know what a Gizmo was, and the UI did little to help her understand.

XA expressed that the application seemed to obviously be in early development and that she did not have much more to say about it.

Expert evaluation by Game Developer (XB)

XB had several years of experience with developing VR games in Unity. She tested a version of the application with Gizmo2 and Rotation1.

She was able to test out the Grabmode as well as Gizmode, and move the dog, scale it so it would fit within the doorframe of the house.

XB said the Grabmode was intuitive and easy to understand, it behaved like she expected.

According to XB it was comfortable to have the Gizmo follow the controller, and she thought the translation was easy to perform. She said however that it felt strange to tap the touchpad to cycle axes, and sometimes she would struggle with finding the right axis.

Rotation1 was a little off to XB, as she noted it would always rotate a little more than expecting, and it was difficult to rotate exactly the way she wanted. She suggested having a combination of Gizmo2 and Rotation2, as they all attempted to rotate her controller in both Rotation1 and Rotation2. She stated this would the most intuitive.

Scaling worked the way XB expected, and she even expressed it was fun to use.

She did however comment that the text at the corner of the screen expressing the selection was not intuitive, but that it should be a highlight instead.

5.1.2 Summary of Notes from Student User Tests

Due to the general spontanity of student visits, it is difficult to say exactly how many students tested the first iteration prototype. My best estimation is somewhere around 20 students. These happened over several weeks with groups of different students and visitors arriving sporadically. Not all students had the time or were willing to sit down for an interview or fill a questionnaire. Instead, the students were instructed to think out loud as notes were taken while observing the students. They were given the same task as the experts. The following is a summary of the notes taken during user tests.

The lack of an interface was the cause of much confusion. Most of what existed of UI would be hard for users to see as they would instinctively hold the controller close to their face, at which point the clipping distance of the Magic Leap will hide any UI appended to the controller. See section 2.4.2 for an explanation on the Clipping Plane. This made many users confused as they thought there was nothing happening at first, and when grabbing objects, they were unsure as to what kind of feedback they were getting in the UI.

The feedback which was most common was that users had difficulties remembering the location and names of different buttons. Most would be able to find the touchpad and home menu, however there were serious struggles with remembering the location and distinction between the trigger and the bumper. To help them, I would make the user push each button while I said the button name, and then have them push each button on their own while reciting the button name. Quickly after this however, the users would often forget where each button was. Additionally, the combination of learning the button functions seemed to confuse the users further.

For most users, the first instinct would be to touch around on the touchpad on the controller. This would not do anything unless an object is selected, and so many users would be confused as to why nothing was happening when they used the touch pad.

Selection was not very clear for most users. A debug text overlay would display which object was selected, but it was difficult for users to connect this to the actions they were performing, claiming that nothing was happening as they selected an object. Users experienced with 3D graphics programs would like to have an outline around the object, as an indicator that the object was selected.

The field of view is smaller than users expect and most of them comment on this being a problem. Sometimes users would get lost in the application, and many would often lose track of Gizmo1. For some of them, having to snap the Gizmo to their controller added a layer of difficulty, and they would rather just always have it visible.

Additionally, many users would quickly get bored after having tested the control scheme. They seemed to be unable to see any kind of potential for creation and interaction in the application, or in any case uninterested in the amount there currently was. After having manipulated an object, many users would say "that's it?" and take off the HMD. Some expressed a want to add their own, or new objects into the scene.

Users Experienced with 3D

Some users experienced with 3D graphics programs expressed comfort around the familiarity of Gizmo1, they however seemed to have issues re-positioning it, and becoming slightly annoyed by the fact it would snap back when letting go of it. Most of them also had issues with scaling, expressing that getting a "precise" scale was difficult. Other than that, as well as a couple errors in the code causing unexpected behaviors, experienced users claimed gizmo1 worked fairly close to their expectations.

Despite some users stating Gizmo1 worked as expected, these same users expressed a preference for gizmo2. One of them stated that it was "much more comfortable" to have the Gizmo attached to the controller as there was no need to keep track of it anymore. Another expressed it to be more intuitive, and less confusing as it felt more like part of the UI. They did however have difficulties with selecting the correct axis because it was confusing to do it by tapping the touchpad. There was no UI indicating which axis was the "next" axis. Still, the impression seemed to be overall positive, with some users stating it was fun to move and scale.

The experienced users also said the Grabmode worked as expected, however most assumed the touchpad was scaling the objects instead of moving it. They would rather have the touchpad scale the object rather than move it, as they want to be able to perform the same transformations as with Gizmode.

Users Inexperienced with 3D

Users inexperienced with 3D graphics programs mostly expressed annoyance with Gizmo1. They struggled to see the difference between the gizmo and other 3D models. This seemed to be because it Most of them seemed to struggle with making a connection between interacting with the Gizmo and the selected object. Often the user would seemingly panic, push all the buttons on the controller and state that nothing was happening. I would attempt to calm them down and explain the controls again, but most users at this point would find it difficult to take in.

After changing to Gizmo2, inexperienced user reactions changed. This time around users would seemingly be less confused about the functionality of the gizmo, most of them stating it was fun to use. However, like the experienced users, they had struggles with selecting the right axis. As a consequence, they also had difficulties performing the intended transformations. They were able to complete the test, some had fewer problems than others, but having the dog be uniformly scaled and rotated correctly was difficult. Sometimes users would get stuck, touching around on the touchpad and not connecting what was happening with the gizmo to their actions. Like the experienced users, they would rotate their controller instinctively when rotating the gizmo in rotation1. Their description of a better version matched a combination of Gizmo2 and Rotation2.

For inexperienced users, Grabmode was the preferred way to transform objects. Most of them would switch to it and not switch back, stating it was "behaving more like I would expect". Like the experienced users, they were confused by the touchpad, thinking it was scaling the object and stating that "the program is scaling it in a weird way". One of them stated that the most intuitive way would be to move it with one finger and to scale it with two. An issue with this is that the touchpad does not allow for multiple fingers.

5.2 Second Iteration User Tests

Before the impact of the Covid-19 pandemic, I was originally planning to do several extensive rounds of user tests with up to 50 visitors of the IMTEL lab. This was right around the corner of the lock down however, and I was only able to do two rounds of user tests with around 15 users in total. Eight of them were able to answer questionnaires. This feedback was used to adjust the application before the final user tests described in chapter 6. In this section I will go through the user tests and the questionnaire results gathered during development of the second iteration.

The test consisted of two users collaboratively making a Co2 particle together. Before this, they were shown a video tutorial described in section 4.4.6. Like earlier user tests, they were instructed to think out loud for us to take notes. Additionally, those who had time were asked to fill out a questionnaire.

5.2.1 Summary of Notes From User Tests

Some users grew bored while watching the tutorial video and would end the experiment before putting on the HMD.

In general, users seemed to be more easily able to understand the Gizmo, and successfully switch between the Gizmo Tool and Grab Tool without getting confused about the difference between the two. Most users would after a couple of minutes get used to the Gizmo and successfully create a Co2 Particle. Some complained however that the Gizmo was difficult to use, and scaling was hard to understand.

Often users would open the menu with the controller too close to their face, hence cutting it off.

Some users also reported during testing that they wanted to touch the menu instead of using the touchpad to navigate it. With the current design that would not be possible as the menu was fixed to the controller. Other users reported being confused by the menu not knowing how to interact with it. Though there were arrows pointing up and down on the touchpad, there was no apparent connection between these and the menu.

Users also noted that when the delete and reset buttons had an object name written on them, they would sometimes clip outside the button, which users said looked ugly.

Selection seemed to be fairly effective for most users, saying they could easily see when an object was selected. However, some users reported a difficulty telling if white objects were selected, as they had the same color as the highlight.

During user testing, users would be greatly confused by the controller layout overlay as they assumed the overlay had a functionality, trying to navigate it or activate it somehow. Some users were also unsure as to which buttons the arrows were pointing at due to drift issues on the device.

5.2.2 Questionnaire

Eight of the participants were able to answer questionnaires about the application.

Only three participants answered whether the tutorial helped them understand the controls or not. Two agreed while one was neutral 5.2.2. Most users understood what the buttons on the controllers did 5.2.2.

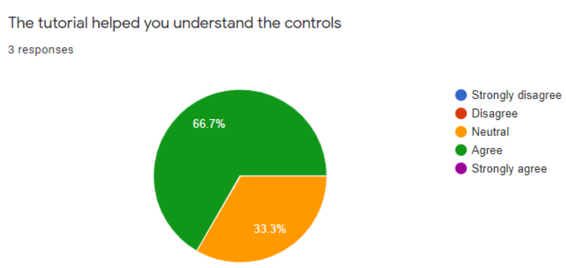

Figure 50: Question 5

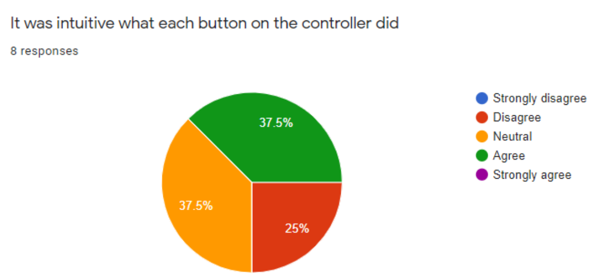

Figure 51: Question 7

Most users agreed that the Gizmo was easy to use except for rotation. Three participants disagreed that rotation was intuitive, three more were neutral while two participants agreed. See figures 5.2.2, 5.2.2. 5.2.2.

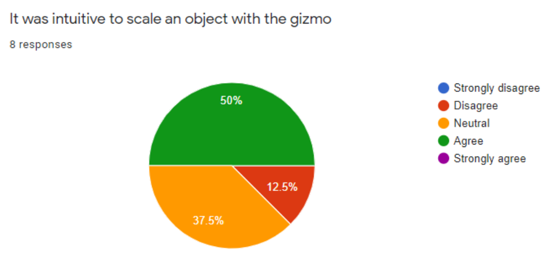

Figure 52: Question 10

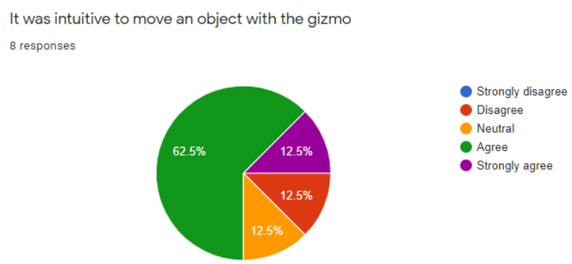

Figure 53: Question 11

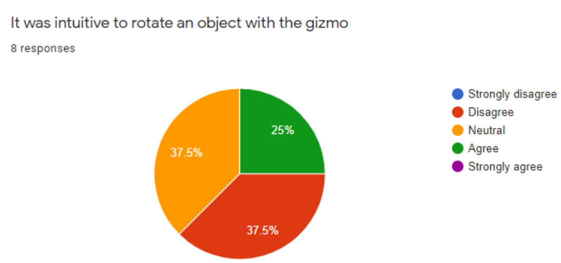

Figure 54: Question 12

Overall, most participants were neutral about manipulating objects with the Gizmo 5.2.2. Two of them would like to not use it at all, though four disagreed 5.2.2. Only one person disagreed that there should be UI text explaining the Gizmo 5.2.2.

Overall participants thought selection was intuitive 5.2.2, 5.2.2, 5.2.2, though only half the participants thought it a selection was clearly marked 5.2.2.

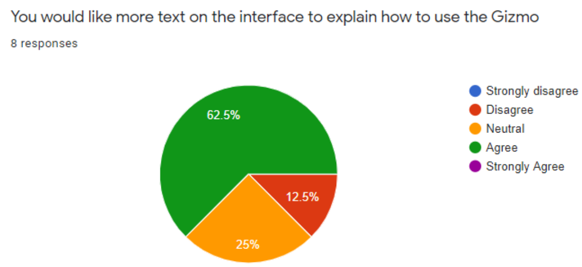

Figure 55: Question 13

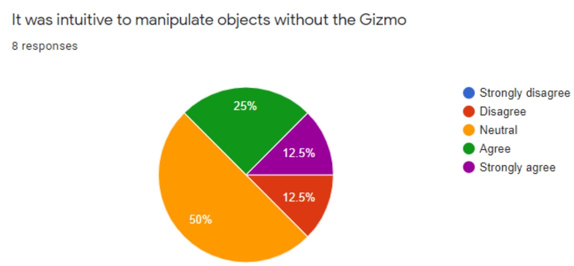

Figure 56: Question 14

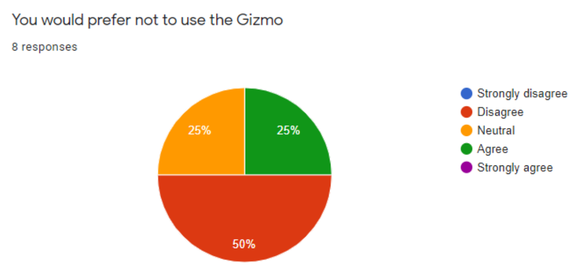

Figure 57: Question 15

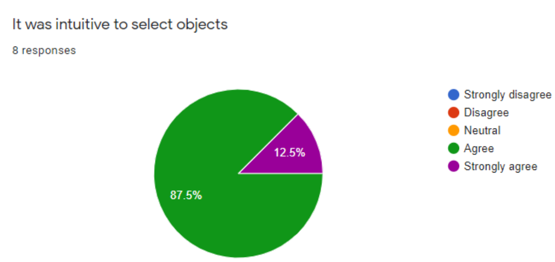

Figure 58: Question 17

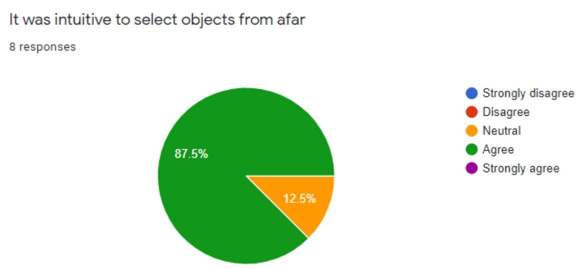

Figure 59: Question 18

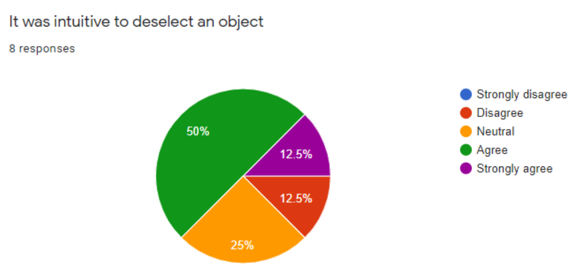

Figure 60: Question 19

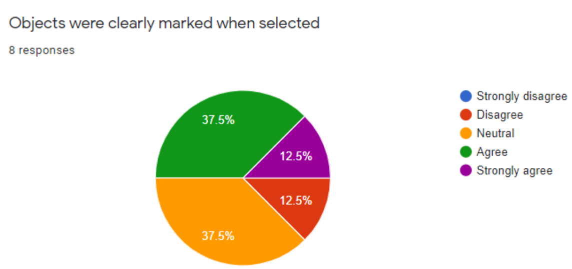

Figure 61: Question 20

It is clear that participants were divisive about the context menu. Half the participants thought the menu was intuitive to open while the rest were either neutral or disagreeing 5.2.2. Three participants disagreed that navigating the menu was intuitive, and only two of them thought it was intuitive 5.2.2.

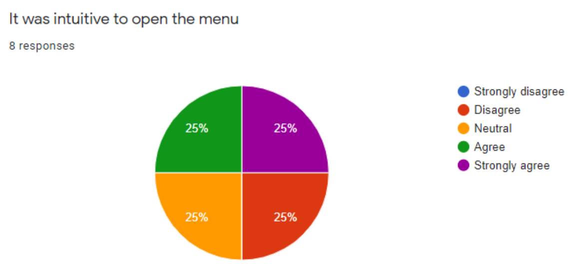

Figure 62: Question 21

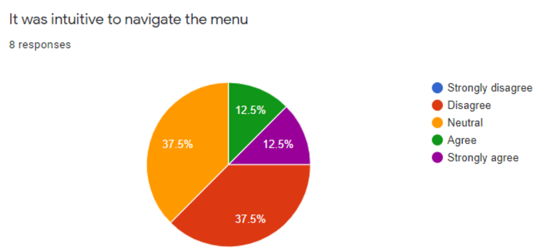

Figure 63: Question 22

6 Final Test Results

This chapter will detail the process and results of interviews and questionnaires evaluating the second iteration of the application, i.e. the final evaluation of the end product. Every participant was asked to watch a demo video of the application beforehand, which is available here:

https://drive.google.com/file/d/1xJdxD55xXNX_GnqIpRBhwHUYJZH6hzJD/view?usp=sharing

For both interviews and questionnaires, the research questions were used as a reference in order to structure them and attempt to ask the right questions.

6.1 Interview Structure

The interviews were conducted over Skype video calls, some of which were time limited. Both the experts and the user test participants were given similar interviews. The differences between the interviews is that the experts were only able to watch the video and were not given system usability questions, and had each interview tailored to them with some questions relating to their field of expertise. The user test participants were all given the same interview questions, and due to them not being time limited I had more time to go through each question.

During interviews, the participants would often bring up new answers to previous questions, or answer questions I had not already asked. Though I did go through a list of questions linearly, it would be confusing to read the answers in this order, so instead I summarized their answers and put them into categories:

- Initial Thoughts on The Application
  Here I asked them to just say whatever immediately came to mind about the application.
- Thoughts on The Interface
  In this section I went through each part of the interface. I asked about the selection system of the different menus and of the 3D objects. I asked what they thought about, and how they would change each of the following:
  - The selection system, both selecting models and selecting buttons in each menu
  - Selection visibility
  - The Gizmo Tool
  - The Grab Tool
  - The Draw Tool
  - The Resource Shelf
  - The 3D avatar
- Thoughts on Cooperation:
  Here I talked in more detail about what they thought about the quality of cooperation, the

different challenges in creating a cooperative application, how to increase cooperation and how to better the 3D avatars specifically for workspace awareness.

- Potential Uses in Different Fields:
  Here I asked them if they thought the application was useful for, and how they would change it to make it more useful for fields such as education, art, and business. I also asked if they could think of other fields.
- Thoughts on The MagicLeap One:
  If I had time, I talked about what they thought about the MagicLeap One, its potential, and future.

6.2 Expert Evaluations

In this section I will go through the interviews with field experts who had been given the video demo of the application beforehand. These interviews were semi-structured, meaning I would often let the subjects talk about what they wanted, but I would also steer the conversation towards my intended questions. The conversations were recorded, and then transcribed.

6.2.1 Expert Evaluation by AR Professor (X1)

This expert had several years of AR development work under his belt and was about to start his job as a professor in the field, with a high focus on performance augmentation. Most interesting to this expert was discussion around UI and how to increase work space awareness.

Initial Thoughts on The Application

X1 found the "clicker" i.e. the controller to be interesting, noting that few other AR HMDs use anything other than hand gestures, but that any that do are low quality. He noted that much of the functionality of the application is like Mixed Reality Toolkit (MRTK), "but on a higher level". The MRTK is a toolkit for cross-platform development of mixed reality application.[58] According to X1 the MRTK is not ready for consumers yet, and that this project could potentially be an alternative if developed further. The manipulation looked similar to MRTK, but the resource shelf looked like something new.

X1 had used similar applications before but noted differences in how users are represented within the application, which I will detail in subsection 6.2.1.

Thoughts on the interface

X1 was interested in the resource shelf. He looked understandable as a menu and noted that some other applications have "shelf like" menus. To them, it seemed intuitive and "easy to grab and interact with". Ideally, X1 noted that the shelf should be dynamic and handle imports of model lists. The shelf should adjust the number of boxes to the number of models in the list, instead of this being defined by the player. When asked for changes to make to the resource shelf, X1 specified that there should be a "preview" version of complex models in the shelf in order to make it less performance heavy.

After moving on to menus and selection, X1 mentioned he thought the radial menu for the Gizmo

Tool was too small but seemed intuitive. Additionally, he emphasized that there needs to be error correction in some way. If someone hits the wrong entry while using the touchpad, it should only preview the action to be applied instead of applying it like it does currently. He continued to say that this needs to be implemented in all areas of the application. When the user selects an object, there needs to be an indication that the object will be selected before the actual selection happens. Additionally, when the user performs a transformation, there should be a way to fine adjust the transformation before applying.

The context menu also seemed to be too small according to X1, and old-fashioned. X1 added that the context menu is a 2D interface in a 3D context, which he thought did not utilize the medium very well. Instead of having buttons, X1 suggested there could be a 3D system where you could grab a tool, or anything that makes the user orient themselves through space to make a choice rather than pushing a button. Buttons are not great "in reality" and therefore do not need to be in augmented reality, X1 added. If using buttons however, X1 noted that gaze tracking could be used as well, to reduce the amount of work done with the hands.

As an additional note, X1 mentions that there ideally should be more libraries to take from in order to not have to build everything from scratch, such as the context menu and Gizmo radial menu. This would make it easier to innovate, as less time would be spent on secondary things like menus.

X1 noted that the object selection system follows the Schneiderman's Mantra, of allowing for finer interaction once zoomed in. This lets the user decide at which level to interact with an object. From a distance, the user can use the selection ray, while up close the user can tap with the cursor directly. He emphasized this as a good way to switch between interaction modes depending on the context. When asked about suggestions for changes in object selection, X1 would make the highlighting on selected objects more noticeable. It should be thicker, and a different color. Ideally the color should be opposite of the selected object. X1 would also add some sort of indicator on the selection ray to tell which surface it is intersecting with. This could for example be a circle that follows the point where the ray and object or physical wall intersects. The ray needs to indicate that it intersects with physical objects so that it does not break the illusion.

Thoughts on cooperation

When asked about cooperation, X1 noted similar applications he had worked with. Applications like Spatial3.1.2 have 3D models of a human torso with a head and arms, which help to convey what the other users are doing. X1 calls this a "social proxy". This is essential for cooperation as it creates a bigger sense of work space awareness. The social proxies are uncanny in Spatial according to X1, because they attempted to imitate too much of the real player while the technology available is not enough to do this convincingly. A picture of the player is overlaid on a 3D model which looks odd, as well as the arm movement being approximated "like puppets". Something more stylized might be more appropriate, such as the cursor, but the cursor itself is not enough. There should also be a head model to indicate gaze. This will help other users see what each other are doing, not by seeing the objects they are moving but seeing it through the social proxy. This could also be conveyed through highlighting an object being looked at. The most important part according to X1 is to make it obvious which objects are being focused on.

Additionally, X1 notes that an in-application voice channel also needs to be present for proper communication.

Potential Uses in Different Fields

In context with education, X1 suggested multiple ideas the application could be tailored to fit, as well as a project he had been working on earlier. The project involved museum artifacts being given back to the original culture they were taken from. There is too big of a risk of damaging the artifacts by moving them, so an alternative is to give them back virtually. This would re-contextualize the artifacts and could be used in an educational manner. Additionally, the application could be tailored to fit a more on-site museum application made for interacting with museum artifacts or taking artifacts "with you" home.

X1 thought an application like this would be particularly useful in a business setting with the current lock down situation. He stated there is potential in the area for new types of spatial computing based remote collaboration. Particularly for training, he elaborated. Experts are rare, expensive, and one would not necessarily expose them to danger. If for example a machine producer needs to fly an expert across the globe to ask for advice, this could be done remotely instead through such an application. "There is a huge potential for these technologies in the business world" X1 said. In order to make it better for business however, X1 said the changes he mentioned about the resource shelf need to be implemented. There needs to be support for online repositories which the resource shelf will automatically pull models from. This in addition to audio support and a higher sense of virtual presence through a social proxy.

When asked about performance augmentation, X1 said the application reminded them of some aspects of what he built or planned to build. In particular the possibility to use superimposed 3D models to guide people through a task. He also said the live aspect is very important. Giving instructions live is expensive, seeing as there needs to be an expert ready at that time. It would be interesting for performance augmentation to look at which aspects of live interaction it is possible to recreate. Having a collaborator asynchronously present to guide the other user. X1 called this a "ghost protocol" and explained it would be more difficult as one would need to find ways to record each movement and put it in context.

6.2.2 Expert Evaluation by Game Development Expert (X2)

This expert has several years of experience as a researcher in AI, gamification, and mobile games. He was very interested in discussing potential ideas and directions to take the application.

Initial Thoughts on The Application

X2 listed off different aspects of the application and his thoughts on them. The resource shelf looked intuitive to X2 in the sense that he would immediately assume the user can do something with it, assuming it as a bookshelf metaphor. His first thought was to pull something out. X2 found an issue with selection however, as he thought it was strange to select an object in order to drag it around, as opposed to selection and drag being the same action. He compared to VR application where "highlighting is immediate grabbing". The context menu seemed too much like a standard UI menu that users would expect. X2 would at least include icons for the buttons.

Thoughts on The Interface

Continuing into the interface part of the interview, X2 wanted to add to the discussion on the context menu. He asked a couple questions about the behavior of the menu and stated he would rather have the menu follow the look direction of the player so that it never gets lost. If not, it should rotate towards the player so that it is never seen from the side. Adding to the statement about icons, X2 was unsure if the average user understands what a Gizmo is, so there would need to be some visual indication on the button. The radial menu on the Gizmo was much more preferred according to X2. Due to the fact that each button has an equal distance from the center, this could be easily navigated with the thumb as well as with a gaze tracker. Gaze tracking might be easier than the touchpad, X2 suggested.

X2 asked if there was a way to modify drawings after making them, which there is not, and commented this would need to be implemented. Deleting, moving, and perhaps duplicating drawings would be useful. Combining objects and drawings into one would also be interesting, he said.

When comparing the gizmo menu and the context menu, X2 pointed out an inconsistency between the two. There is an obvious difference between the way users interact with the gizmo menu, it being radial. Seeing as X2 preferred the radial menu, he would change the context menu to be radial as well and increase consistency. He also added that the context menu icons could be arrows for the Gizmo, the cursor for the Grab tool, and a pencil for the draw tool.

To improve the usability of the Gizmo tool, X2 would add more functionality. There is a reset button in the context menu, but X2 would like to see a reset button for each axis and transform, i.e. "reset x scale" or "reset y rotation". Otherwise, he said the tool seemed fine.

X2 asked if the operations in the context menu were object specific, and added that this could be useful. For example, if an object had animations, there could be buttons to start and stop animations. If an object had a high quality and a low quality version, there could be buttons to change between them. Additionally, X2 would prefer to add coloring options to the context menu, or options to change materials.

Thoughts on Cooperation

X2 thought an interesting addition to the application would be if users were able to collaboratively create "new" objects together. This way there would be more possibilities to create. For example, in the video demo, I put together a coronavirus model. X2 suggested that there could be some way to enter an "object creation space" with other players, where the creation in that space becomes a new object once every player leaves. Either that, or some way to select multiple objects and put them back in the resource shelf, registering them as a new object.

Regarding the visualization of the other players, X2 seemed to misunderstand the player cursor as an icon to indicate player position. He commented that a simple 3D model to indicate their torso and head position could me more useful. I would clarify further into the interview that the cursor indicates the position and rotation of the player controller, which he found to be more useful than just the player position. However, a 3D model of the head would still be much more helpful.

On top of that, X2 added that visualizing which tool the other users are using would be helpful.

X2 offered an example of a feature which could improve cooperation. This feature is something he had experience with from a VR application, where users were able to "hand over" tools to each other. This means he could throw objects to each other, hand them over, and both of them could grab the object at the same time. X2 said this made it feel like the other users were present, despite being in different rooms.

He added that different objects in the environment could provide higher degrees of interactivity between users. This could for example be screens which could display content from a user's computer or show a video from a streaming site. Additional objects could depend on the context, for example in an educational setting, objects could react with each other. A plant could grow, and flint could spark fire.

Potential Uses in Different Fields

There could be a lot of application cases in education, according to X2. Especially in educating engineering. The application could be expanded to add interaction and simulation between objects. For example, elements could be put together to create chemicals, or objects could be put together to create machines. This could be implemented through a building block system. Physics could be added to see how many beams a bridge would need to hold a car, or how much weight a train could pull. X2 explains it as an "exploration environment" where users could experiment with putting objects together to see if the result is stable enough to fill a need or fulfill a purpose. The basic tools in the application could be the same X2 said, but the objects in the resource shelf would be more sophisticated. These would have to be tailored to fit the specific field.

What immediately came to mind for X2 when asked about business was an "IKEA solution". By this he meant having the ability to decorate a room with items from a catalog. Here he would integrate the "object specific functions" he mentioned earlier so that a user could swap between versions of furniture. This could mean changing the object colors and materials, and also swapping between sized of furniture.

Additionally, X2 imagined the application being used to put together factories and seeing whether he would work or not. This included seeing whether a door was big enough for the machines or vehicles to pass through, and whether the conveyor belt setup would fit within the enclosure etc. If there were reactions between objects such as in the educational setting, there could be ways to test the machinery, for example.

The application could be useful for anything art related according to X2. Given functionality to re-mesh any object created within the application, it could then be processed for 3D printing. There are a lot of tools which could help make that easier, so X2 seemed confident it could be done.

Lastly X2 imagined there would be a use for the application in the video game industry. For example, putting together environments or constructing objects to place in the environments. According to X2 it could have a lot of inherent value.

6.2.3 Comments on MagicLeap

Additionally, X2 had a short discussion about the MagicLeap One. He had used it multiple times but said he had not used it enough to have any opinions on it. However, there were some advantages X2 immediately saw with the MagicLeap One which revolved around the controller. Since most AR HMDs do not use controllers, he are dependent on tracking the hand position. Since the field of view on most these headsets is small, the tracking is often lost when the user turns their head. This issue is nullified with a controller which is always tracked, such as with the MagicLeap One. The MagicLeap One will track the controller position so long as it is within range. Since the user does not need to have their hands in front of their face as much, their arms do not get as tired, and it might end up feeling much more natural than without a controller, according to X2. "If you're running in a video game, using a controller or a keyboard. If it is well implemented, after a while it just feels like you're running, you don't think about the controller" he explained.

6.2.4 Expert Evaluation by Artist (X3)

This artist was currently finishing a PhD, specializing in innovative design. Mostly she wanted to talk about alternate, more attention-grabbing ways to implement the interface.

Initial Thoughts on The Application

X3 initially talked about how the current state of meetings having to go through applications like Zoom because of the lockdown, is excessive. They happen on flat screens that hurt to look at and "recreate reality in a terrible way". This technology is not designed to be used as excessively as it is now, and is not pleasurable, she elaborated.

His Initial Thoughts on The Application were that it was very coherent and intuitive, and she liked having the elements in the application suspended in the air. Having objects suspended in the air makes it easier to handle them as opposed to if she had to be on a surface. X3 stated however that she thought it was almost too coherent and intuitive, and it didn't surprise them enough. There should ideally be more content that she "didn't know was useful", or in other words things that were new and more innovative.

Still, X3 thought it was interesting to see an application like this in AR, as she does not lose a sense of self as much as in VR.

Thoughts on The Interface

X3 described the Gizmo tool as intuitive, but more so for people who have experience with video games and controllers for video games. For people who have not been acquainted with a joystick for example, and this is still most of the world, she elaborated. This is because the navigation of the radial menu for the Gizmo requires the user to use a touchpad, which she estimated is about the same difficulty as a joystick. X3 said there are other ways of interacting with menus that could be more intuitive for people without a video game background, but it is something she can still learn relatively easy. Regarding the other tools, X3 said they were "cool" and that she would like to see more of them.

When asked about ways to potentially improve the tools, X3 said she would not change them.

The tools in the application seemed useful, as they could be used to place objects on her desk in the office instead of having them there physically. Something she would like to see, would be more gesture handling. X3 described wanting to hug, pet, and grab objects with her hands.

As X3 mentioned earlier, she liked having objects suspended in the air, but she would also like to see different ways to handle the resource shelf rather than having boxes lined up like a shelf. Recreations of reality are boring to X3, and so a shelf is not an exciting metaphor. Instead, she suggested a series of other ideas, such as: A toy box, a tree which grows object on the branches, or an animal that walks around with objects on its back. She also described an application she had worked on where the user would lay down, and the interface would spin around the user. "Nobody wants to be in a recreation of their office space", X3 put it. Additionally, X3 mentioned that having all the boxes in the resource shelf be the same size conveys that they are all the same importance, which could be misleading. Again X3 listed more ideas such as a conveyor, or a Möbius strip, or an assistant with six appendages. These ideas are less constricted by the idea of a work environment and physical world physics and would utilize the medium more according to X3.

Thoughts on Cooperation

When asked about usefulness for cooperation, X3 had difficulties answering. She did not seem to be very interested in discussing the cooperation aspect, however the following is the series of comments she gave.

X3 stated that she disliked the term useful and that an artist does not want something useful, but rather something generative or inspiring. She said however, that something which would be useful for this kind of application is the ability to simulate certain kinds of emotions in the environment rather than replicating reality. She elaborated that using visualizations to help communicate ideas in fields such as philosophy and education. For example, assigning abstract shapes to concepts in philosophy class and putting them in a context.

After being asked to elaborate a bit more, X3 explained that there is no way to fully learn how to be a surgeon without actually cutting into a body. However, there could be ways to emulate parts of the surgeon environment and see for example the conditions it takes to make the user sweat or become stressed. This would require simulating a situational environment in as much detail as possible by any means necessary.

Potential Uses in Different Fields

X3 was mostly interested in the application being used in an educational setting. The current state of education is too boring according to X3, and any means of making it more exciting would be seen as a plus. Any way to learn something while pretending to be in a different setting, or seeing something other than a lecture hall, and especially in the current lock down setting.

6.3 Evaluations Through User Tests

User tests were conducted with an additional student and I, as well as a test subject. The test subject would be made familiar with the application by experimenting and asking questions, as well as building something with the other student once they felt comfortable enough. After this, a semi structured interview was held.

Some things which are not visible in the interviews is that the users would often end up using the Grab tool for moving objects fast and easy, then moving on to the Gizmo tool once they had it in the approximate position for fine adjusting and scaling. Some would talk about potential ideas for the application during the experiment and would explain using the draw tool. It should also be said that due to there not being a way to remove drawings, some users were afraid to use it, resulting in them not using it in explanations.

6.3.1 User Test Subject 1 (U1)

This user is the one who had been helping conduct the user test experiments, and had the most experience with the application. He also has experience developing on the Hololens 2.

Initial Thoughts on The Application

U1 thought the application overall felt natural. Having used it extensively however, he discovered a few bugs and had difficulties performing certain actions that were more complicated.

Thoughts on The Interface

After having used touch screens for years, he did not feel like the interface was too big of a change. The other menus on the MagicLeap One control similarly to the menus in the application, using the triggers to select etc. He thought using the touchpad on the controller for selecting an axis on the Gizmo was a little strange, but he got used to it, saying it still felt intuitive. U1 added that people without experience with the MagicLeap One might struggle a little with anything requiring use of the touchpad however, as it is not immediately apparent that UI attached to the touchpad means it can be manipulated by it.

U1 found several issues with the Gizmo tool. One issue was scaling. U1 struggled with understanding how to maneuver the controller in order to scale something down. At first, he thought he could only increase scale, rather than decrease it. U1 said that a visual indicator could help for this, to show that the Gizmo tool itself is a pivot point, and that it uses the distance to the controller as input. There was also an issue he noticed where he had difficulties understanding which direction the Gizmo would transform something. When he scaled on the Z axis, it would sometimes scale in a different direction than he expected. To fix this, U1 would move the Gizmo tool from the controller and to the object instead and copying its local axis coordinations.

U1 also thought that there could be more indicators letting the user know which objects he was about to select. For example a highlight that happens once the selection ray intersects with the object, for it then to be highlighted differently once it is selected.

He also mentioned that not everyone will know what a Gizmo tool is, so the menu needs to have some sort of indicator of what a gizmo does before the user selects it. For example, a visual indicator like an icon of arrows.

Using the grab tool, the user is able to scale objects with the touchpad, but seeing as there is no visual indicator on the touchpad that this is possible, U1 did not know about that initially. He found out eventually but would have appreciated for example an arrow or something similar. Another issue with the scaling he noticed, is that an object can be scaled negatively with the Grab tool, but not with the Gizmo tool. This confused U1.

U1 thought the affordance of the draw tool was low, because there seemed to be no visual indicator of what it does before drawing with it. He did not notice the visual distinction between the Grab tool and the Draw tool, and thought he had done something wrong initially. However, the usability was high, as it was immediately easy to use once he started.

The Gizmo tool was slightly more difficult to use for U1, who preferred to use the Grab tool for quick manipulations. However, he pointed out that there are a lot more operations available with the Gizmo. Despite it being more difficult to use, it is still very useful according to U1. He would like some changes in behavior on it however. For example, both the Gizmo and Grab tool could have a function which allows users to move things away or closer in a quicker manner. U1's suggestion was to make objects move quicker the further away they are, or that they had momentum and could be thrown.

At first U1 thought the resource shelf was just a gallery, and only understood it as a menu once he dragged an object out. This was different from the "normal" approach of a simple scrollable 2D list which the user could drag objects out from. U1 did not see why this could not be a tab in the context menu. Other than that, U1 thought it worked fine, but would like to be able to re-position the resource shelf like with other objects. Additionally, he would like to discard the idea of having a shared resource shelf, and instead have his own, invisible to other players.

In general, U1 would add more feedback to the application by adding more sounds and vibrations when performing operations. For example when selecting an item on the Gizmo radial menu, the controller would vibrate. When duplicating an object, it would make a sound, etc. "The more senses you appeal to the better", though U1 specified it is important to not overdo it.

Thoughts on Cooperation

Cooperation in the application felt natural for U1, and he reported not having any issues. What he would prefer to have is a 3D model representing the other player's heads and torso. U1 felt just seeing the cursor was not enough as it was too small and could easily get hidden.

One change U1 would make is to add a differently colored highlight so objects that are selected by other players. To differentiate between different players, U1 would give assign them different highlight colors. Sometimes U1 would struggle with other users while manipulating an object, because both users would have the object selected, and he did not see until both were moving it.

U1 mentioned working asynchronously in the application could be interesting, but he would want to see more options in the application in that case. Features such as saving and loading, discarding save files, etc. He also thought the usefulness of asynchronous cooperation would depend on the context and field of use.

Potential Uses in Different Fields

U1 said He absolutely thought an application like this would be useful in an educational setting. He imagined an example where users could gather around a table with 3D models and discuss changes to make. For example with an oil platform, he could draw to highlight locations to adjust, sketch out a new heli-platform etc. U1 also imagined another scenario in science class, where there could be 3D models of the sun, the earth, grass, and animals. This could be used to illustrate for example the food chain U1 continued.

If U1 were able to take the application in any direction he wanted, he would develop it in one of two directions. The first is an application where users collaborate on building a 3D model. To increase the usefulness of the application, he would add features such as hiding and un-hiding parts of a 3D model and cutting a 3D model to look inside. Features which would give users more control over how the model is seen and share this across the application. This could be useful in fields like architecture and engineering. The second is something which could be more steered towards the entertainment industry or education for younger children. An application which focuses more on building blocks. U1 compared this to Minecraft, a video game focusing on building with cubes. Incidentally, Minecraft has also implemented features meant to teach kids coding. [59]

Thoughts on The MagicLeap One

U1, having experience with development on the MagicLeap One, noted how he at first thought developing on it was boring and "weirdly put together". After a couple of months of development however, U1 places it equal with the Hololens 2, saying he enjoy working with it now.

There are some aspects of the MagicLeap One that U1 dislikes, still. The device sits clumsily on the head and feels weird to put on and take off compared to the Hololens 2. There are also many tools provided with the Hololens 2 given it comes with the Mixed Reality Toolkit mentioned by X1 in section 6.2.1. This provides, amongst other tools, tools for performance and debugging, which has not been able to find an equivalent for in the MagicLeap One.

One clear advantage of the MagicLeap One according to U1 is the controller which provides precise location and rotation input for the software. Other HMDs use hands for gesture recognition, which U1 suggests might be better in the long term. Some gestures can feel very natural, but others are not, and can be difficult for users to get used to. The controller can make things simpler for the user and the programmer.

Another advantage is the Light Pack which allows for the HMD to handle more complex software with higher polygon amounts, but U1 has not noticed enough of a difference for it to matter.

U1 thinks that the application would work on the Hololens 2 given that the gesture recognition is much better and could allow for precise interaction with 3D models. Some functionality in the application would be more intuitive with the use of gestures, such as moving, rotating, and scaling. But at the same time, this comes with multiple physical constraints, which would not be good for advanced users according to U1.

U1 continues to say that the controller is best for drawing, as well as the more complex functionalities, like the Gizmo tool. UI in combination with gestures is still very new and difficult to do right. The MagicLeap One has buttons, which should be used.

6.3.2 User Test Subject 2 (U2)

U2 had been developing with the MagicLeap One for approximately a year, and had experience developing video games in Unreal Engine.

Thoughts on The Interface

U2 commented he liked the Gizmo tool a lot and that the interface was easy to use. An issue he had was that having it constricted to the controller would sometimes make it difficult for them to see as the cut-off distance for rendering on the MagicLeap One made them have to stretch out his arms a lot. He also had issues with tapping objects with the cursor while the Gizmo tool was active, as the cursor itself becomes hidden. When precision was in issue in selection, that became a big problem for U2, though the selection ray worked fine.

Selection could have been more visible for U2. He suggested making the highlight bright orange and giving it a glow effect. With the white outline, it was difficult for U2 to tell if white objects were selected or not. If not orange, U2 would choose a color not often used.

To U2, there did not seem to be any visual indicator of which transform mode the Gizmo was in. He would only know by memory. To fix it he would add some more visual indications on which mode the Gizmo was in, like a highlight on the radial menu.

U2 felt like the draw tool worked well, but that it should have an eraser or a delete option.

On the usability scale of one to ten, he would put the Gizmo at six to seven. The Move tool was an eight or a nine, while the draw tool was an eight. The context menu he rated nine. He did not think any of the tools were too complicated, and he would not need help from a technical person. People who have played video games before should understand easily. However, he did think object selection was more complicated than needed. He commented how selection with a mouse is usually on the down-press, while selection in the application happens once the user lets go of the trigger. This could be difficult for users without a video game background.

Thoughts on Cooperation

Overall, he felt like there was not enough indication of where in space the users were. An arrow pointing towards where they were, would be preferred. The field of view is small on the MagicLeap One, so instead of looking around for other users he would like a visual indicator.

In addition to that, U2 felt like a representation of the other users' headset is needed to convey their gaze and give them more sense of presence. He would also add names above the user heads, to make it easier to distinguish between users.

U2 experienced that what I had in my application was in the middle of the room, while for them it was in their wall. What U2 would like is for there to be indication where other users' walls are. This way it could be easier to agree on where in space to build. Ideally the application should automatically sync the rooms of each user.

Thoughts on The MagicLeap One

U2 felt like the application would not work on any other AR HMDs, as it relies on the controller. As far as U2 knows, there are no other AR HMDs with controllers. The application would have to be rewritten significantly if it were to work on a different HMD. However, U2 worries about the future of ML as a company as well as the difficulty of working with the MagicLeap One. There are news of the company not paying rent for their office, and in his experience development for the MagicLeap One is difficult. The help forums and communities are largely inactive and secretive, U2 says.

6.3.3 User Test Subject 3 (U3)

U3 was at the time of the interview writing his thesis with the MagicLeap One, though were attempting to switch over to the Hololens. He had over a year experience with the MagicLeap One, Unreal Engine and Unity Engine.

Thoughts on The Interface

U3 thought the highlighting was intuitive, and the radial menu increasing when interacting with it worked well. He did however think the objects were a bit big. The resource shelf itself could have been a smaller size. This in combination with most the menus being too small for U3 to read easily without glasses. The visual distinction between the different transform modes on the Gizmo made it easy for U3 to see what the intended use was. It seemed to U3 that he was supposed to be able to see which tool the other users were using, but he could not tell what it was representing.

If U3 wanted to change anything about selection, he would change the highlighting to make it bigger and more distinct. Other than that U3 said he thought it was easy to understand. The radial menu made sense and was easy to select with. He did not notice it was possible to tap objects to select but thought that the selection ray worked fine.

He would change the Gizmo tool by highlighting the active buttons in the radial menu. While using the application he would sometimes forget which buttons were active.

For the Grab tool, U3 seemed to find there to be an inconsistency between how he held the controller and how he expected the object to move. He would also like for there to be a button to push so that an object would go further away or closer. To increase the feedback, a visual indication of an object being grabbed such as a change in color on the cursor would be nice, according to U3.

When resetting an object, it seemed inconsistent how big objects would scale, because they were not always the same size as the object he pulled out of the resource shelf.

U3 would not change the Draw tool other than adding a color picker and a delete function.

Overall U3 would like to add more feedback to the application, such as haptic feedback when selecting something or highlighting a button, or sounds when objects are being duplicated, moved or deleted. More indications that the user is doing something right. the ML controller also has LEDs which can be used, as U3 suggested.

U3 felt like the tools were fairly user-friendly and could not think of any changes he would make, other than adding a button overlay to make it easier for newer users.

Thoughts on Cooperation

In general, U3 thought cooperation went smoothly and was fairly easy. He could see other users moving objects around, and he could take objects from them and vice versa. Given the field of view, he would still like something pointing to other users' position. Additionally, he would like an indicator of what tools other users are using.

U3 commented that 3D avatars are "great" but that just the cursor was sufficient for the type of tasks he had to perform. He also commented that if two users are in the same room then it would not matter as he can just look at each other in the physical world.

Potential Uses in Different Fields

U3 pointed out that although the MagicLeap One is arguably state-of-the-art technology, it is still fairly limited, and the field of view is small. He said if it were to be used in an educational setting, there would have to be "a bunch of workarounds" to make users more comfortable with the field of view. U3 imagined a scenario where students sat around a table collaboratively working on a model, for example something with human anatomy, and that the field of view would not be an issue in this case. A similar setting would be if everyone had a replicated view of a model on their own desk, instead of staring at something from afar.

For business, U3 could imagine it being used in maintenance and construction. He mentioned however that this could be tricky as the application does not seem to be designed for it. Everything would have to be redesigned to be tailored to the specific business. He mentioned multi-object selection as something that would be essential for business, other than that the main functions could stay the same.

If he could take the application in any direction he wanted, U3 would focus on creating something for museum interaction. Specifically mentioning an application made for building puzzling in AR for kids in museum settings.

Thoughts on The MagicLeap One

U3 starts off saying the MagicLeap One is the one people should develop for, out of current HMDs available. The Hololens field of view is not as good and their gesture recognition is limited. He added however, that he do not recommend anyone to buy an ML over a Hololens now, as he reported the MagicLeap One being rather difficult to develop for. The gesture recognition on the MagicLeap One is worse than the Hololens, and often crashes according to U3. U3 has experience with the Hololens 2, saying it will most likely be better to develop for, given the bigger community and improved gesture tracking. According to U3, no other HMDs seem to be comparable.

6.3.4 User Test Subject 4 (U4)

This user had some experience developing with the MagicLeap One but had mainly used other HMDs.

Thoughts on The Interface

U4 was fairly brief in his feedback. He reported the selection ray as easy to use but sometimes selection with the touchpad on the context menu seemed to be faulty, adding that he would "debug" to make it more useful.

He found the Gizmo useful and easy to use, selecting on the radial Gizmo menu to be a bit difficult. There was no haptic or sound feedback when a button was selected, so U4 had issues knowing if he had accidentally selected something or not. He also noticed an issue where it is possible to invert an object with the Grab tool, but not with the Gizmo tool. U4 would add vibration when buttons are selected and make it possible to combine axes and transformation modes in the Gizmo tool.

In comparison to the Gizmo tool, U4 said the Grab tool was simple and easy to use but had fewer features. In that sense the Gizmo had more uses.

U4 had no issues with the draw tool, stating it as easy to use. If he were to change it, he would add a color picker.

There were also no problems with the resource shelf, it was easy to use and understandable. Still, U4 would rather use a 2D scrollable menu fixed to the controller. He did not see a reason to have it fixed in space.

Thoughts on Cooperation

U4 reported he felt a sense of the other users' gaze was missing and suggested either showing the selection ray from other players or casting a ray of their eye gaze.

The cursor was too small for U4's liking, as it would easily get lost in between other models.

Additionally, U4 added that the application could be strengthened by a lower artificiality and using more aspects from the physical world. Otherwise, the application might as well be in VR instead.

Potential Uses in Different Fields

Education wise, U4 could only see this application being useful for higher education levels, for example in design. He could not see it being useful for children's education. He thought the same about business, that it could be useful for discussing large designs. For art, the only change U4 would make is adding more color, an eraser, and brush sizes. He would also attempt to make drawings look smoother for other players.

Thoughts on The MagicLeap One

U4 did not feel like he had enough experience with the MagicLeap One to comment much on the HMD itself. However, he did comment that this application could be more useful in VR if it were to be remote. In a remote setting it is no different from VR according to U4. Any other AR HMDs would not fit the application as it would be annoying to perform the same operations with gestures. Especially with how horrible gesture recognition is now, U4 added. It could only work if gesture and voice recognition were improved greatly.

6.3.5 User Test Subject 5 (U5)

This user had no prior experience with AR other than Pokemon GO, but some experience with VR. He had never heard of MagicLeap One before.

Thoughts on The Interface

Overall U5 saw the interface as natural, but struggled a bit with the axes on the gizmo. The direction an object would move was not the expected direction after rotating it. Most things felt natural, and he had no problems with them, but not the Gizmo tool. The Grab tool was fine, except when he scaled something to the point where he inverted it. Lowering the scale would then make the object "bigger" while increasing the scale would make it "smaller", which confused U5.

Selection worked the way U5 would expect, and he thought it was easy to see when he had selected something, except he was missing a function where the user could snap an object close if it was far away, through either a button or a gesture. He would also like something that would make it easier to work on objects from afar.

When asked if he would make any changes to selection, U5 said he would not change the behavior, but he would add something. Specifically, he would add the ability to select drawing objects. He would also add the ability to select and move multiple objects at the same time. This could be done with a lasso-like tool or a 3D box selection tool.

The context menu seemed fine for U5; however he was a bit confused reading the labels on the buttons. Buttons like "Draw tool" and "Grab tool" were self-explanatory, but he had no idea what to expect from the "Gizmo tool" button. Some more info would be nice for U5, in the form of for example a short description.

While speaking of the Gizmo tool, U5 mentioned the radial menu felt too sensitive. He then continued to describe the touchpad scaling on the Grab tool, most likely having mistakenly swapped their names. U5 had issues scaling with the Grab tool, as whether he increased or decreased scale is dependent on touching the upper or lower part of the touchpad. U5 said he got better at it, but it was still difficult to tell which part was the center of the touchpad, saying it seemed wrong. There were no icons to guide where to push on the touchpad, so U5 thought something like that would help. For example, arrows indicating where to push for scaling up and scaling down. Having to see where his thumb was in relation to the controller was difficult, as he had to "separate from the virtual world" which was mentally taxing. This was simpler with the Gizmo tool as there was more UI giving the user a point of reference for where to move the controller.

Choosing the correct axis on the Gizmo tool was not always easy, U5 said. The active axis displayed on the tool itself seemed to mismatch with the axis being transformed on the selected object. For example, after turning an object 90 degrees on the Z axis, the X and Y axis would be pointed a different direction, but there was no indication of this on the object.

U5 was unsure what he would do to change the Gizmo tool, he did not want to add too many buttons on it, as there were many already. He was wondering if it would be possible to add a gripping function to the Gizmo, where gripping the controller harder would let the user grab an object.

One thing U5 thought was missing was the ability to color objects and suggested adding a "paint" function to the Draw tool. This would work by assigning a color to the paint tool, drawing on an object, and that object then adopting the selected color. Ideally this would work per mesh within an object as well, meaning users could paint each leg on the oil rig model a different color.

Continuing with the draw tool, U5 did not expect while drawing that the lines would be ribbon shaped. He expected them to be more like tubes and suggested changing them to that.

Moving on to the resource shelf, U5 said he easily understood what it was for. However, some objects were small, or had simple shapes and could have been small. Small objects do not need as big of a box, and therefore the Resource Shelf could have "sub boxes" where one box was turned into four, to save space. Still, U5 thought it was easy to use, and compared it to menus from the video game "Garry’s Mod".

Thoughts on Cooperation

When asked about collaboration, U5 said he had no difficulties cooperating, but had issues reading visual cues. Also, when turned away from the other user, he had no idea what they were doing as there were no audio cues either. When U5 was actively looking at the other users, he had no problem understanding each other, but when he looked away, he could easily lose each other or get confused.

U5 would like to differentiate between the other users more, and suggested adding multiple colors to the other users. Assigning every user their own cursor color. In addition, he suggested having list of users and their colors, which could be moved around in the scene and placed on the wall. U5 suggested this would make it easier to tell which user is which.

Sometimes U5 and other users would select the same object and try to adjust it. U5 did not understand what was going on until he asked the other user if they were also attempting to adjust the object. To prevent this, U5 suggested adding highlighting on objects selected by other users. The highlight would ideally be the same color as the user. When attempting to select something which is already selected, U5 would like to get feedback telling them he cannot.

For U5, the resource shelf was in the middle of the room. But for other collaborators, it was in their wall. This could make it difficult to cooperate sometimes as he would have to move the objects to a place where both could work. Multi object selection could make it easier to adjust the scene for when more players join and the objects are beyond their walls.

Potential Uses in Different Fields

Being able to open up models and separate them would help a lot to make the application useful for education according to U5. The ability to hide and un-hide layers of a model as well. This could for example be used in explaining the human anatomy.

For art, U5 said it could be useful if there were importing and exporting possibilities. Being able to save something to work on it later. Also, functionality for flattening a drawing so it can be exported for a 2D program.

U5 says that for business, the cut-off distance would have to be smaller because users will want to look closer at objects. For businesses like architecture, adding functionality to help draw straighter lines would help. But he can imagine architects using it just for visualizing a project.

When asked if he could imagine applications like these being used in a pandemic setting, U5 said he could only imagine it for work. When confined to his room like he currently was, he would rather play VR video games with a VR HMD rather than using AR applications with an AR HMD.

6.3.6 User Test Subject 6 (U6)

This user had no experience with VR or AR and had never heard of MagicLeap One.

Thoughts on The Interface

U6 thought selection with the selection beam was fine, but needed some time getting used to it. He said he was much more used to just using a mouse and keyboard and that it felt weird using a "laser pointer". He did not feel like using the tap to select function as he could do what he wanted from a distance.

To U6, it was easy to see when objects were selected, and he did not feel like it needed to be changed.

The context menu seemed intuitive to U6, although he thought it was annoying that there would always be an active button when opening the menu. This meant that if U6 opened the menu and pushed either the trigger or the bumper, that button would be activated despite U6 not intending to push it. U6 was also unsure which buttons to push to use the menu. Something that helps the user see which buttons to push, and not having an active button when opening the menu would be better for U6.

U6 thought the Gizmo tool was intuitive, and he understood how to use the radial menu as well as how the tool itself functioned. He did however not notice that there was a visual difference between the transformation modes. When asked whether he would change the Gizmo or add anything, he said no.

When asked about the Grab tool however, he started talking about having to pull more than he expected in order to move it, and that it was not 1:1. I asked if he was confusing the Grab tool with the Gizmo Move mode, to which he answered yes. He understood what it was doing but said he felt like he had to pull too much, but only in move mode.

It was intimidating to draw at first, according to U6. There was no clear indication where the line would be placed, so he would have to draw first to understand how to use the tool. He was also confused at first because he had to move the tool at a higher speed than he preferred to in order to place a line. If he were to change it, he would add a color picker, and lower the speed threshold for drawing.

The resource shelf was fine, according to U6. He did not understand immediately, but once he pulled out an object, he got it.

Thoughts on Cooperation

U6 said cooperation was easy in the application as far as he experienced. He did however have trouble understanding what other users were doing other than seeing the results of what he had done. Preferably, he would see more visual indicators of what users were doing on their cursor. He would also make the cursor more visible, as he sometimes would lose track of where other players were.

Potential Uses in Different Fields

U6 could imagine applications like these in educational settings, explaining some courses need illustrations which could be made more practical through AR. Architecture and design for example, could be done remotely instead of students having to meet up in class. There could be internet courses held in it. In order to make it more fit for educational settings however, U6 says the application would have to be easier to use so there is less of a barrier to use it.

U6 did not have much to say about the art field, other than he thought the draw tool would need to be more sensitive.

For business, U6 said the application would be a great way to present new ideas for projects, adding that it has a nice visual component.

In a pandemic lock down setting, U6 thinks applications like this would be useful.

6.4 Questionnaires

In addition to interviews, there were digital questionnaires sent out to the IMTEL Slack as well as social media. These were answered by 24 people. As can be seen in figure 64, most the answers were from males, most of them were between 20 and 40 years, and most of them did not have experience with 3D modeling programs. In figure 65 we see that most of them have experience with video games and are familiar with both AR and VR. Only four people were familiar with MagicLeap One.

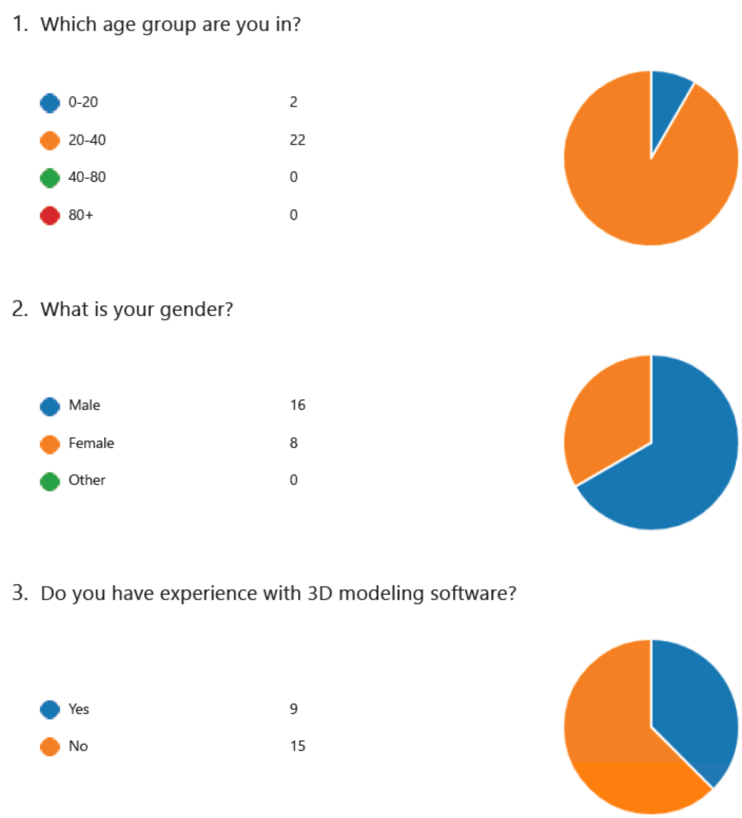

Figure 64: Questions one through three

6.4.1 Object Selection

Most participants either strongly agreed or agreed that selection with the selection ray and tapping appeared intuitive. Only one person disagreed that the selection ray was intuitive, as can be seen in figure 66 It seems however that most participants were struggling to see whether an object was

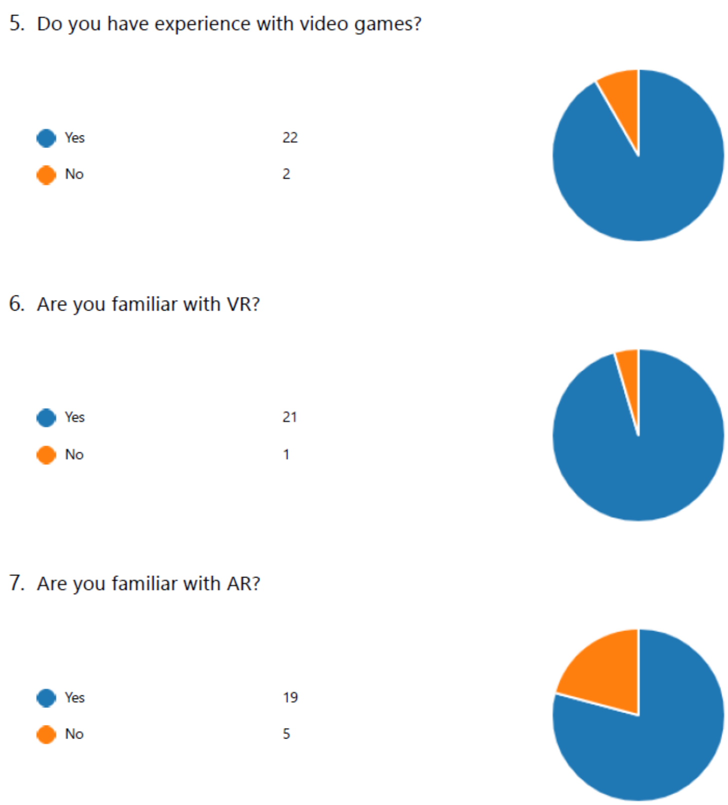

Figure 65: Questions five through seven

selected according to figure 67. In table 10.2 we can see the suggestions from participants. Most suggestions revolve around making the highlight thicker or a different color, which coincides with the results from the interviews as well.

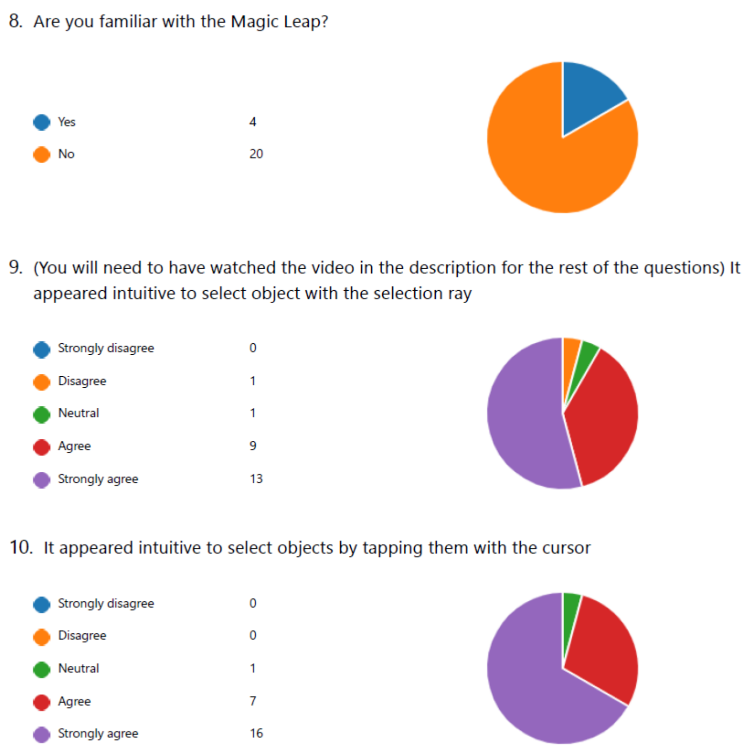

Figure 66: Questions eight through ten

6.4.2 Questions About The Context Menu

Like with object selection, we see in figure 68 that most participants strongly agreed or agreed that buttons were intuitive to select and were visibly selected. One participant strongly disagreed, however. We can see in table 10.2 that most the suggestions of improving the menu vary greatly, but the selected tool being displayed differently is mentioned multiple times.

6.4.3 The Tools

According to figure 69, most participants strongly agreed or agreed that the Gizmo menu was intuitive, and it was easy to tell which transform mode was active. Written feedback was again quite varied, but some form of extra instructions was mentioned twice. See 10.2

Feedback on the usage of the Gizmo tool itself was fairly positive as well, most users strongly agreeing that it appeared intuitive as we can see in figure 70 Written feedback mostly mentioned not making changes, but adding a visual cue that the tool is grabbing was mentioned multiple

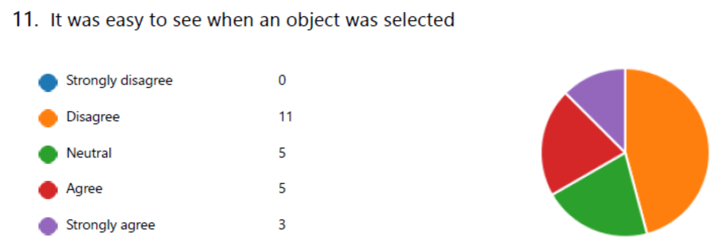

Figure 67: Question 11

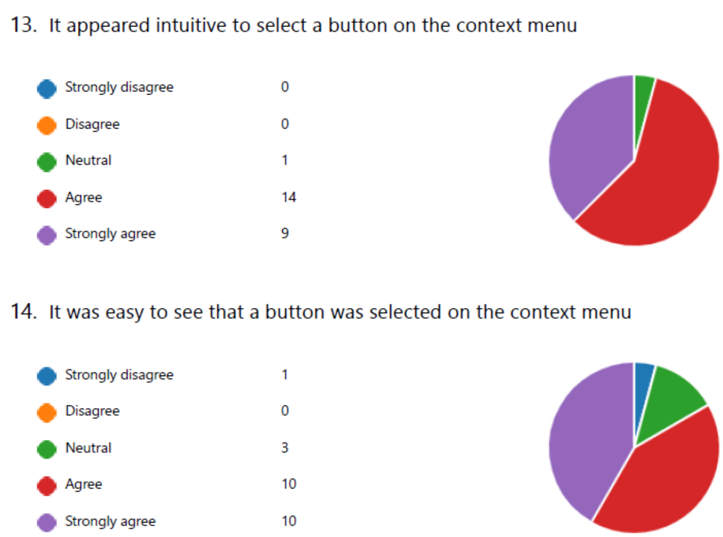

Figure 68: Questions 13 and 14

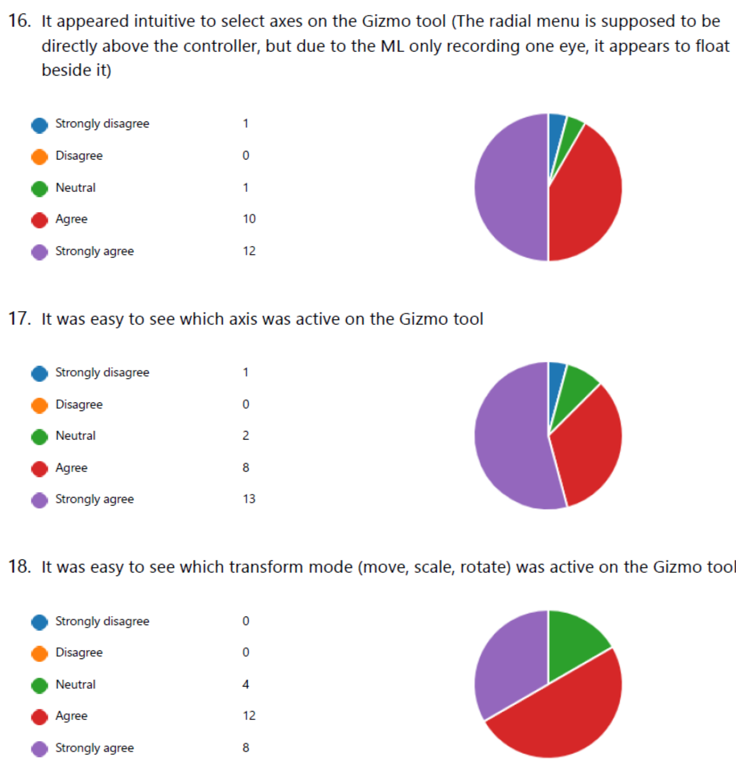

Figure 69: Questions 16-18

times. See 10.2 for more.

Most users strongly agreed that the Draw tool appeared intuitive 70 though the written feedback shows participants thought the lines needed to be smoother for other players. For more feedback, see 10.2

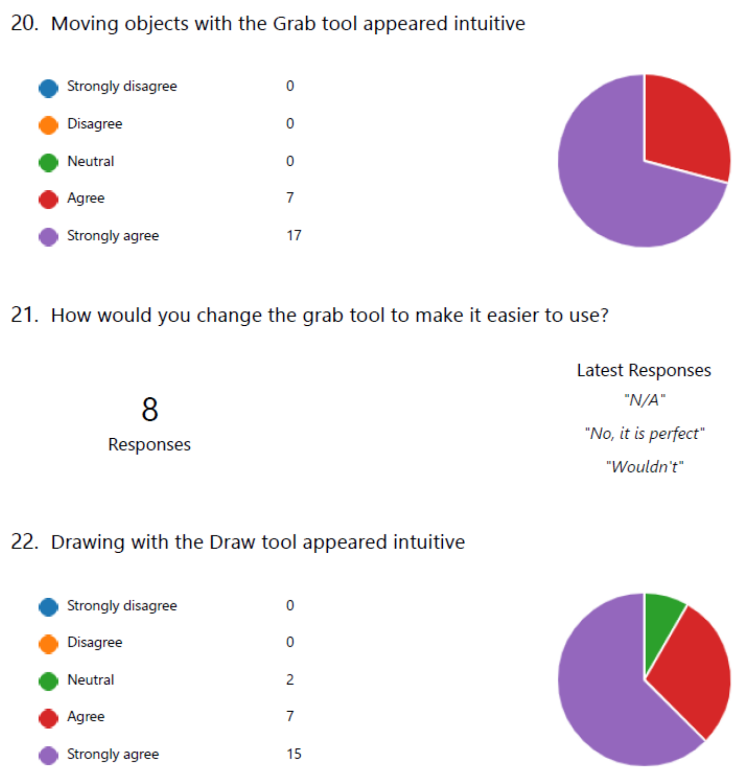

Figure 70: Questions 20-22

6.4.4 Questions about Cooperation

Most users agreed that cooperation appeared easy and that it was easy to see what other players were doing, as we can see in figure 71 and 72. However, it is clear that the participants are missing visual cues to tie other users to the objects they are manipulating, as well as more visualization of the other users. Participants suggest different colors for other users, 3D models for their heads, and being able to see the tools they are using. This also coincides with the results from interviews. See 10.2 and 10.2]for details.

6.4.5 Usability

As can be seen in figure 73, most users agreed that the application looked easy to use, and there were few inconsistencies according to the participants. They do not all agree, but most either agree

Figure 71: Question 24 and 25

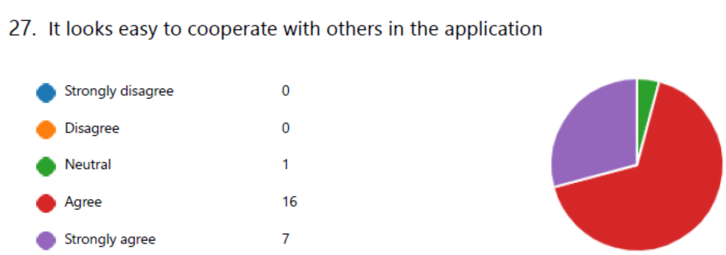

Figure 72: Question 27

or are neutral about using the application frequently.

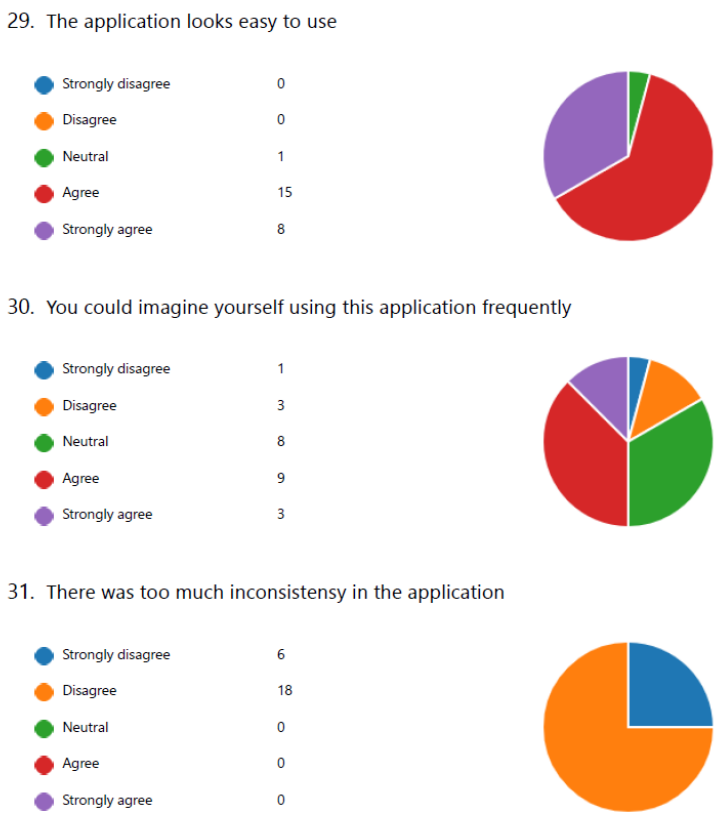

Figure 73: Questions 29-31

6.4.6 Fields of Use

It seems most participants strongly agree that the application could have use in an educational context. Of the courses they thought would be useful, the most voted for are Science, Arts and Crafts, Medicine, and architecture. These are also courses which were mentioned in the interviews. Suggestions to make it more useful for education involve animation and a higher degree of interactivity, as well as ability to leave notes. See figure 74 and 10.2 for more details.

Participants agree that the application could be useful in art as well. Most written feedback focuses on adding more drawing and 3D manipulation tools. Specifically, the ability to modify 3D models is mentioned multiple times. See 75 and 10.2.

Lastly, most users also agree that the application could be useful in business context. In 77 we see that the most voted for businesses were architecture, construction, city planning, and engineering. Several of which were mentioned in interviews. Written feedback on how to make it more useful

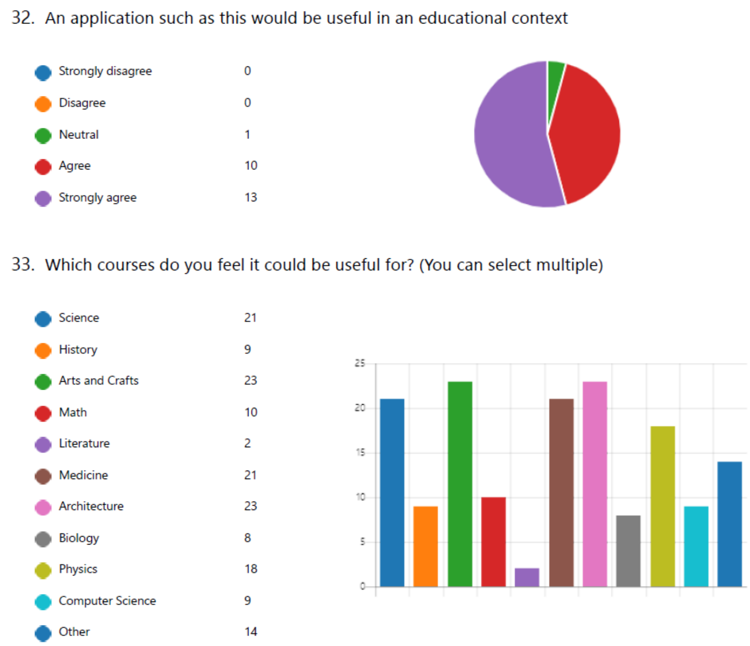

Figure 74: Question 32 and 33

varies. See 76 and 10.2

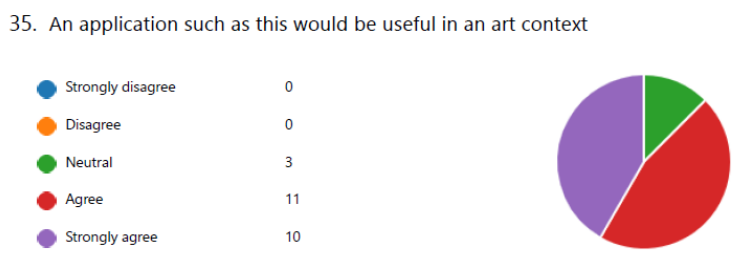

Figure 75: Question 35

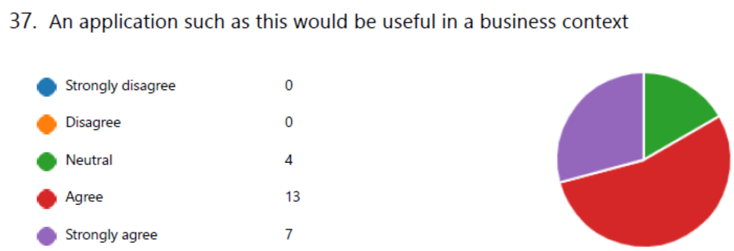

Figure 76: Question 37

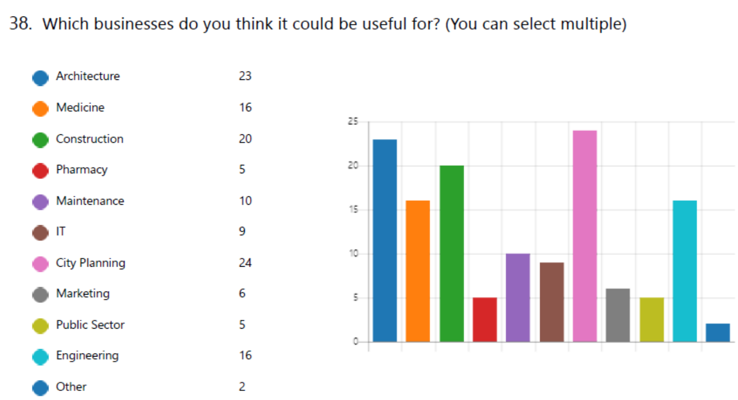

Figure 77: Question 38

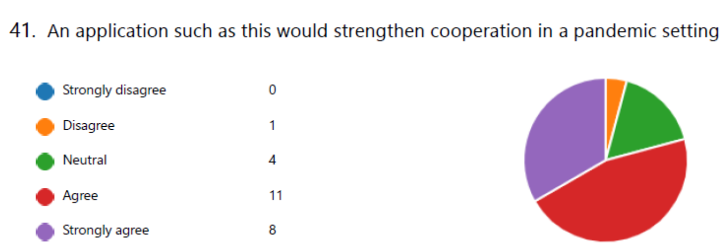

Figure 78: Question 41

7 Discussion

7.1 Discussion of The MagicLeap

Early development on the MagicLeap One was easy and straightforward as there are multiple tutorials on their developer websites on how to make a starter project in Unity Engine, Unreal Engine, and their native language Lumin Runtime [60]. Getting initial prototypes built was also easy and straight forward thanks to the "Zero Iteration". Zero Iteration runs the project locally on the computer, but streams information about the position and rotation of the headset as well as controller to the computer. This way projects can be run without building on the HMD. Building is time-consuming, so this allows for faster development and debugging [61]. This can be done by connecting the HMD to the computer via a USB hub which takes an additional power cable in order to charge the HMD while transferring data. Zero Iteration is launched by another software called "The Lab" which acts as a hub for other ML related software, such as the device bridge which allows users to download files from the HMD, or the package manager.

Despite this however, the MagicLeap One is not perfect, and neither are the development resources. Often The Lab will not recognize the MagicLeap One, which I did not find any solution for other than restarting both the HMD and the computer. There are heavy and unpredictable drift issues which vary between devices. This means that UI attached to the controller will often drift away. Sometimes more than 5 cm from the controller, rendering it useless if it is supposed to be near a button or the touch pad. Gesture recognition is very poor on the MagicLeap, and will sometimes not even run. The headset will often lose track of its position in space and will enter a reset mode which stops the application for about 10 seconds. This also happens in rooms that are "dimly lit" according to the device, which in reality is not always the case. From interviews with other developers, there are also multiple manufacture errors with MagicLeap Ones, such as lenses being tinted with green. The clipping plane on the MagicLeap One cuts off anything within 37 cm, which every test subject pointed out without fail. The first assumption is that this is a mistake made on my end, but it is in fact a deliberate decision made by the MagicLeap developers.

The MagicLeap One has a geolocation restriction which does not allow anyone outside certain specified countries to download applications from the internet. This means that several applications which could have been useful for testing and comparison were unavailable.

The MagicLeap developers release updates for the software regularly, which sends a warning to the users to update within a certain time limit, or they will be cut off from the internet. Over the winter, one of IMTEL's MagicLeap Ones sat without being updated. This resulted in the MagicLeap One not being usable until it was factory reset.

To assist with development, the MagicLeap development team has developed multiple packages which can be downloaded and imported into the Unity project. However, these packages seem to be both in the early stages and outdated at the same time. Some packages would give me over 200 warning messages in the Unity console after importing because of outdated code. For the first half year of development of this application, there were no packages for networking or any form of sophisticated object interaction. There were packages for UI but only to a small extent. User interfaces and object interaction would have to be created from scratch, slowing development down significantly. In the second half of the year spent developing, the MagicLeap development team launched a new iteration of their development kit including new object interaction tools as well as networking tools which could make it possible to sync HMDs with each other. They also underwent a change in marketing and branding which focused more on corporations [62], re-branding and re-releasing their package manager as well as Zero Iteration. This meant that their previous package manager was obsolete and not available for use, and neither was the Zero Iteration. With the new package manager not being released yet, this meant the packages were unavailable as well as it not being possible to test without building. Once the packages became available, the guide on their websites was incomplete, which required me to contact developers. This took several weeks and affected development. In the end the packages ended up being dropped as the same time could be spent developing instead.

Not all software will run on the MagicLeap. There are libraries such as the ARToolkit [63] but there is little information on how to get it running with a MagicLeap One. There are similar issues with Photon, which I will discuss in section 7.2. Contacting the MagicLeap development team, they instructed me to go to their forums for development questions. In recent years, activity on their forums has died down and I would only get answers from the MagicLeap development team on my questions after no one else answered. Using the MagicLeap developer forums is frustrating due to their requirement of using email confirmation code every time a user enters the website.

7.2 Discussion of Development

The end product was programmed with C# in Unity, using Photon Engine for multiplayer functionality and MagicLeap packages to make the project compatible with the MagicLeap One.

Overall, working with Photon was easy as the Photon forums are very active. However, There were multiple issues where Photon would cause Unity to crash ften. Whenever this happened during building, it would corrupt the entire project and I would have to manually delete corrupt folders in order to open the project at all. This problem would vary between computers. On some computers it would often happen, and on others very seldom. It was very difficult to see a pattern, and so I could not figure out what to report.

During the Covid-19 lockdown, I was unable to take the MagicLeap One anywhere. This was due to the MagicLeap One being expensive and fragile, as well as it being a difficult and lenghty process to order a new one. This meant I had to stay in Trondheim and were not able to travel home, which caused a great deal of stress and slowed down development. I was, however, still able to implement many of the intended features.

7.3 Photon Engine

There are no tutorials on how to get a MagicLeap One application working with Photon, only with PUN voice 2. Support for MagicLeap One multiplayer applications is in other words low, and as it would turn out during development, PUN voice requires developers to be a member of a closed circle. I would only find this out after digging around in install requirements of an add-on listed in the required packages for PUN voice 2, meaning a lot of time was lost.

7.4 Lack of Resources

Due to a lack of developer resources, many elements would have to be coded from scratch. This meant that a lot of time had to be dedicated to coding non-essential elements of the application, such as control schemes for menu interaction, and the menus themselves. The application was not designed to be menu-centred, but the menu turned out during user tests to be a big barrier for users. If there were more resources available, more time could have been spent developing the Gizmo and Grab tool, Draw tool, etc. In the end there was not enough time to complete as many tasks as intended.

7.4.1 Re-prioritization

Despite technical issues, perhaps one of the biggest issues in retrospect was the re-prioritization of features due to Covid-19. It happened halfway into development, and much of the application had to be refocused, which ended up affecting development schedules greatly. I had already done this multiple times due to taking user feedback as input for prioritizing tasks, but it became more difficult to handle because of Covid-19. Tasks had to be re-prioritized, added and removed unpredictably. An issue that was known from early on was that the 3D head model used as a 3D avatar was not visible when running the application. The 3D model was a child of the MagicLeap Unity camera, which seemed to become hidden when running the application. Fixing it required studying the code behind the MagicLeap camera, which was noted as a non-essential task. The new prioritizing was based on fixing and polishing features and functionality which were already there, or nearly there. Regrettably, this affected the 3D avatar which was then cause of a lot of complaints in the feedback.

7.4.2 Tutorial Video

A clear issue with the tutorial video was that it was too slow moving and too long for users to pay attention. In one-on-one testing scenarios where I was able to take it slowly and give my attention to the user, they would often be able to pay attention and understand. However in scenarios with more users at a time, they often lost interest during the tutorial video and walked off before trying the application.

7.5 Discussion of Results

Results were also greatly affected by the Covid-19 outbreak. Due to the lock down, I was not able to acquire students to test at the IMTEL lab and were unable to observe interaction in as great a detail as intended. Much of the time which could have been spent developing, was instead spent attempting to get in contact with test subjects, waiting for cooperators, and re-scheduling. Most the data were gathered by sending experts and test participants a video recorded on the MagicLeap One. The MagicLeap One's camera records 4:3, 1440x1080 resolution. The footage is grainy and not as clear as it seems when using the HMD. The footage also cuts off some of what the user sees while recording, which makes it difficult to know which parts will be within frame. This caused some elements to look less visible for test participants who only saw the application through video.

However, I would like to point out that despite all the difficulties, I was still able to successfully test my application and demonstrate the cooperative potential it had to an extent. I was still able to gather feedback from over 30 people, 6 of which were able to test the application with a MagicLeap One. This is more than I initially expected given the situation.

As we can see in both the interviews and questionnaires, a great portion of the feedback was centered around the weak highlighting, which users who had been able to test the application were more positive about. They did however agree that the color could have been different.

After having spent a significant portion of development time developing the context menu, there was still a lot of feedback saying it was difficult to read, not aesthetically pleasing, and not innovative as a menu. Despite the context menu not being a central part of the application, it still formed a barrier for test participants. This again affected feedback on other tools, which mainly were that they needed to be further developed, have more functions and overall be more sophisticated.

We can also see from the feedback in the questionnaires and interviews that the 3D avatar was not as detailed nor sophisticated as it ideally should have. This was already something I knew was important from studying existing works in 3 and the literature review in section 2.2.1. A more detailed 3D avatar would add more work space awareness in a remote setting. But was originally deprioritized because of the initial focus on co-located cooperation. Again, because the physical person would be in place of the 3D avatar. Like Beck explained in section 3.3.1, a digital recreation will never be as good as the actual physical presence and there will still be confusions. Once The project had to be re-focused however, it may have been wrongly re-prioritized with the time constraints in mind, as a 3D avatar is still better than nothing.

There was not only negative feedback, however. Many users found aspects such as the Gizmo and the Grab Tool to be intuitive. I was not able to document it, but several participants messaged us over social media saying it was an impressive application that looked fun to use. During user tests, participants would laugh and joke with each other. This is something which could potentially negatively affect the educational aspect as it disctracts the students. It does however, increase their enthusiasm to learn the application itself. Sometimes testers would request to be allowed to stay in the application for longer because they thought it was fun to use.

7.6 Discussion of Requirements

First I will discuss the functional requirements, then I will discuss the non-functional requirements. I am going to be looking at whether I felt like they were met, and why.

7.6.1 Functional Requirements

The UI needs to mitigate the impact of the limited field of view and the unfamiliarity of the MagicLeap One controller

As mentioned in section 4.4.3, there was an attempt made at a UI overlay meant to visualize the functionality of each button on the controller, which ended up being scrapped due to drift issues in the tracking. Without this overlay, there are some elements which help familiarize the user with the controller, but not many. The radial menu on the Gizmo is attached to the touchpad and shares the same shape as the touchpad. The effect of this is that the user is encouraged to touch the touchpad, thus familiarizing themselves with part of the controller. Originally there was also supposed to be a tutorial video which explains each part of the controller in detail. This was also scrapped, however, due to time constraints. For these reasons, it is safe to say this requirement was not met.

Manipulate 3D objects

The user can manipulate objects by performing transformations such as scaling, rotating, translating, deleting, duplicating, and resetting scale and rotation.

Multiplayer

Multiplayer was implemented using the Photon Engine. There is no limit to how many users can play simultaneously. It can however be discussed how successfully the multiplayer was implemented. Due to time constraints, no real handling of users quitting the application was written. This meant that users quitting the application would cause other users' applications to crash.

Spatial Alignment

Spatial alignment would turn out to be too difficult and too low priority after the Covid-19 breakout, resulting in it not being implemented.

Add and delete 3D objects

Users can add objects to the scene by dragging them from the resource shelf, or duplicating them in the context menu. Users can also delete objects from the context menu.

Select objects from up close and afar

Objects can be selected from afar with the selection ray by holding the bumper down, and they can be selected from up close by tapping the bumper while touching them.

Gesture and point at objects

Users can convey hand movements with the controller, and use the direction of the cursor to point. This could be improved however, by adding a head model to give more context to the hand gestures, and enabling other users to see the selection ray. Gestures which users can perform at this point are limited.

Hide and un-hide objects

Due to time constraints, this was another feature which was not implemented.

Select objects from a list

This was yet another feature not implemented due to time constraints.

Draw in 3D space

Users can draw in 3D space with the draw tool. There is however no way to adjust the drawing after placing a line. Due to time constraints, I was not able to implement a color picker, eraser, or a way to move drawings around.

Voice communication

As explained in section 4.4.5, voice chat ended up being too time-consuming to implement, as it involved becoming a member of a closed developer group. Voice chat was not implemented.

7.6.2 Non-functional Requirements

The UI should avoid relying on memory

This was fulfilled to some extent. The second iteration relied less on memory than the first, as the Gizmo was attached to the controller it was obvious to see whether the user was using the Gizmo Tool or not as several test participants said in chapter 6. User tests revealed however that there were still some aspects which relied on memory. Like U3 said in section 6.3.3, highlighting the active button on the Gizmo radial menu could help users see more clearly which state the Gizmo is in. Because users overall had difficulties distinguishing the transformation modes on the Gizmo visually, they would often just rely on memory to know which mode they were in. The same issue persists with the distinction between the Draw tool and the Grab tool. The visual distinction between the two is that instead of a ball and a cone, the cursor becomes a cylinder and a cone. The cylinder in the Draw tool is clearly too small for the distinction to be made, resulting in users sometimes forgetting which mode they are in.

Operations need to be easy to perform

User feedback indicates that some aspects of the application were sufficiently simple while some were needlessly complicated. Both expert evaluations and user evaluations mentioned selection being "two part" as too complicated. Some interviewees did not mention this at all. At the same time, there were some complaints in the questionnaires that the UI was not informative enough, several participants suggesting adding more visible information. This could be interpreted as the UI being too simple, or that it in any case was not efficient at conveying information.

Someone who has never user the application before should be able to use it themselves

Multiple test subjects had tried the application without my instructions and reported it to be fine after some experimentation. Most the feedback in the questionnaires was also positive that they could use the application without assistance.

Holograms must stay in place with minimal drift unless moved by the user

Overall, the MagicLeap One was consistent with keeping objects still in 3D space. The problems with drift reported in section 7.1 were not as noticeable on the objects themselves, more so the UI. There were however issues with the MagicLeap One losing headpose, which would then reorient the digital content in physical space once the headpose was regained.

The hardware must be available for purchase

As of now, the MagicLeap One can be bought at: https://shop.magicleap.com/#/

The operating system of the hardware must be continuously updated so that it is compatible with the latest development platforms

The MagicLeap development team continuously sends updates to MagicLeap Ones connected to the internet.

The Gizmo needs to be nonthreatening to new users

According to the questionnaire evaluations, the Gizmo was easy to understand. While performing experiments, users seemed to have a good understanding of the Gizmo after attempting to manipulate an object with it. It is clear from the interviews however that the Gizmo did not always perform the way users expected, which would contribute to them often choosing the Grab tool for quicker transformations.

7.7 Discussion of Research Questions

Here I will describe the possible answers to the research questions I defined in section 1.4.

7.7.1 Main RQ: How to support collaborative work on 3D content in an educational setting with AR/MR?

I learned about multiple aspects of AR collaborative work during the development of this project. During development and user testing, I discovered that social proxies are not very important so long as users are in the same room. Even without spatial alignment or voice chat, users were still able to look at each other and communicate despite confusion about the location and orientation of objects being collaborated on. When not able to point with their physical fingers, they were able to point with the AR cursor. This amount of work space awareness is not ideal and would be greatly improved with spatial alignment. This would mean that users would be able to point and gesture directly at AR objects without confusion, which is essential in an educational setting. The most brought up settings were architecture and medicine, where being able to point at points of interest is essential.

When users are remote, the spatial alignment does not matter, as their physical work environments are not the same. However, at this point the 3D avatar becomes a problem. With users not being there physically, there needs to be more in place of their body language. Voice chat becomes even more essential as well. In order to perform the remote experiments successfully, I used voice chat applications on my mobile phones during the experiments.

It is also clear that there needs to be more functionality to an application meant specifically for education. As we see in the feedback, users were missing more annotation options, the ability to gesture more precisely and see other users' gaze more clearly.

7.7.2 RQ1: How to develop collaborative applications on the MagicLeap One?

What are the challenges of connecting several MagicLeap Ones?

Multiplayer applications with the MagicLeap One went surprisingly smoothly with Photon. As mentioned in section 7.2, Photon and MagicLeap One works quite well despite it clearly not being written with the MagicLeap One in mind. Issues that persist are more with the medium itself instead of the MagicLeap One as a device. Syncing the position and orientation of the AR world in the physical world across devices requires making use of physical world data. It is possible to sync the scanned environment between MagicLeap Ones but only on devices on the same network. Therefore, writing an application which does this while simultaneously including users from different networks requires more sophisticated coding than what I was able to do with my time constraints. The easiest way to do this is most likely to use image tracking which uses a tracked image from the physical world as an anchor for the AR world as discussed in section 4.4.4

Additionally, voice chat needs to be implemented. This requires different frameworks than with VR because everything runs locally on the MagicLeap One. Using the Photon engine for voice chat on the MagicLeap One requires an add-on which, as opposed to the rest of PUN voice is not open source and requires a membership in their developer circle. There are other options however, such as Unity Dissonance, a real time voice chat system which is compatible with MagicLeap One [64]. I was, however, not able to test it out.

After starting user testing with the multiplayer version of the application, it was immediately noticeable that there was a potential, as users immediately showed enthusiasm. Single player users would get bored quickly, but once there was another player involved, they would start playing with the tools, exploring, and communicating.

What are the useful additions MagicLeap One adds to communication?

With the MagicLeap's ability to share the world anchor with other MagicLeaps on the same network through spatial alignment (see 4.4.4), it is possible for MagicLeap One applications to potentially strengthen communication by adding AR objects which exist in the same location for all users. With spatial alignment, users could for example see each other drawing in space around them, which is of course not possible without AR HMDs. This makes it much easier to convey ideas as I found in the user test evaluations. Spatial Alignment is software specifically written for MagicLeap One, however given the Hololens uses comparable technology it is easy to imagine the Hololens having similar software soon if not already.

Discussion can be made on whether the MagicLeap One controller adds to communication or constricts it. Like X2 discussed in section 6.2.3, a controller can end up feeling more natural than without one. In the specific example of this application, users indicated that the application would be annoying to use without a controller. Performing complex operations like scaling on a specific axis is most likely to become difficult with gestures as 6.3.1 mentions. This could be an advantage the MagicLeap One holds over other HMDs, being the only HMD with a controller. However, if the application were to be centered around communication through hand gestures, the MagicLeap One would be a poor choice given the poor gesture recognition.

7.7.3 RQ2: Does the Gizmo affect the affordances of 3D manipulation?

Would a Gizmo be more useful than a gesture-based grabbing system?

As we can see from the interview results, most the transform operations can be performed with gestures: grabbing something and thus inherently rotating it, and scaling. However, performing more complex transformations such as translation on a locked axis, requires a more UI heavy approach. There could potentially be different ways to represent the axes, but the Gizmo is something which already exists, and inexperienced users easily understand it according to the interview results.

What are the advantages and disadvantages of using the Gizmo in this context?

One disadvantage is that most users are new to the Gizmo, especially in the context of AR, which adds a cognitive load, potentially slowing down learning. A clear advantage is that once users are used to the Gizmo, they understand that it can be easier to use and make more precise manipulations.

In general, gizmos of different kinds could be tailored to be more beneficial to educational different educational contexts. The Gizmo as it is now might be too tailored strictly for 3D modeling principles. If we step outside our current definition of Gizmo and go back to the general definition mentioned in 2.7.1, in that it is a 3D manipulation tool represented with 3D graphics, the design and implementation of the Gizmo can change drastically between scenarios.

Perhaps several Gizmos could be developed which could be more useful for educational settings. Gizmos made specifically for adjusting and moving body parts for science class, or Gizmos specifically tailored for engine maintenance in a remote expert setting.

How to make the Gizmo as easy to understand as possible in AR/MR?

Firstly it needs to be very obvious that the Gizmo is not a part of the environment, it needs to be immediately recognized as UI. One way we found to do this is to constrict the Gizmo to the controller. Secondly the shape of the Gizmo needs to be more recognizable than the one I made. As multiple interviews mentioned 6.3, it was difficult to tell the shape of the arrows. This shape is what distinguishes each transform mode from each other, so it should be much more obvious. Thirdly, specifically in the context of educational context, it is likely that the Gizmo will be new to most users, and they need to be aided in some way. For example motion graphics displaying how to interact with the Gizmo, or short explanations when hovering over different parts of it. Different suggestions on how to make it more easily understandable can be found in table 10.2

7.7.4 RQ3: How to support collaborative work on 3D educational content?

What are the minimum levels of features needed for a collaborative application?

Most the minimum level features I have discovered so far are implemented in the application. Ideally, there should be more features than this, but these are what I found to be the most important.

UI wise there needs to be 3D representations of the other users, and a representation for the "self" as well. These need to be representative of the user's body language in some way, such as being able to convey gestures. There needs to be interaction controls which allows for users to cooperate, such as 3D manipulation of objects. This will allow users to "hand" objects over to each other, and collaboratively build something.

Functionality wise there needs to be communication other than visual. Either in the shape of text chat, or a voice chat. This will speed up communication, and without it, communication will be stunted.

What are the affordances of 3D manipulation in single-player vs multiplayer?

Comparing the user experiences in user tests in the first and second iteration, I discovered that users seem to immediately see more of what they can do with the application once there are multiple users in it. Users were able to see each other work, and seemed to want to show each other what they were doing, which increased their motivation and may have led to more exploration, seeing more uses in the application. Gibson's (1986, p127) [65] explanation of affordance as the relation between an animal and the environment is quite clearly illustrated as users see what the application offers differently when they use it in single-player as opposed to multiplayer. As mentioned in 5.1.2, users trying the single player application had issues seeing what the application could be used for or what to do in it, while users playing with each other would often exclaim while playing, the different uses the application could have.

In what educational contexts is collaborative AR/MR useful?

We have seen from this project that AR has the potential to strengthen learning and collaboration, but it depends heavily on the software and the quality of said software. I cannot say that this application in is enough to strengthen learning and collaboration, as it is clearly still in an early stage of development. I can say however that MR has fewer barriers to strengthen communication given that it can make use of the physical users and their body language, as opposed to VR which must rebuild it from scratch. In strengthening communication, it will also by extension strengthen learning. In the case of remote users however, this advantage is lost.

To see which specific educational contexts would be useful, we can look to expert feedback, user feedback, and questionnaires. Experts such as X1 discussed the potential to tailor the application for performance augmentation, meaning it could be used for remote, asynchronous learning 6.2.1. X2 discussed a specific use case for engineering which could most likely be tailored for lower level education such as high school chemistry. A version of the application which implements interaction between objects, resulting in for example chemical reactions when objects intersect 6.2.2.

Users such as U1 talked about the possibilities of using the application in science class, showcasing for example the food chain 6.3.1. U3 also described what appeared to be science class, with students picking apart a human anatomy model together 6.3.3. U6 discussed the possibility of using it for architecture, presumably university level 6.3.6.

As discussed in section 6.4 the most popular choices for courses which applications like this could be useful are architecture, arts and crafts, science, and medicine.

What we can pull from this is that the application as it is now seems to be fit for use within courses which require observation and understanding of physical objects and elements. In section 2.5 we see research which supports this, that AR can strengthen the understanding of physical space and relations.

Which industries can benefit from AR/MR?

I learned in the expert interviews in section 6.2 that there is a lot of potential for collaborative applications within multiple different fields. The most relevant are construction and maintenance, where there is need for remote experts to inspect and comment. They could always benefit from a more dynamic interaction system with the virtual world. Another example is the construction setting where users might need to double check if everything fits in their warehouse, through their door, etc. There are solutions out there but none of them are ideal yet and there is still much potential.

7.8 Discussion of The Six Dimensions

In section 2.2.1 I described the six dimensions of an AR application, and in this section, I am going to discuss which categories my application falls into.

7.8.1 Time

My application is synchronous, because at the moment it is not possible to save or load scenes. It is also not possible to record sessions. Currently, for two users to collaborate in my application, they need to both be on the application at the same time.

7.8.2 Space

Although I was only able to get the final evaluations on remote collaboration, the application supports both colocated and remote work. It might be that the application is less confusing when all users are remote, because they do not experience the difference between the location and orientation of virtual space for each user. This difference is more noticeable when users are colocated. However, colocated users might still be able to communicate more successfully as they can see each other's physical bodies.

Combining space and time together and looking at figure 6, it might fit better into the "Face to face interactions" square as it could be easier for users who are sharing the same space due to the lack of messaging systems or voice chats.

7.8.3 Symmetry

Currently, the application is symmetrical as there are no different roles and every user has the same number of tools and privileges available. If the application is developed further with remote experts in mind, or teacher-student relationships, it may need different roles. As it stands however, there is little reason to add less symmetry to the application.

7.8.4 Artificiality

Currently, the application has a high degree of artificiality, as the application makes little use of data from the physical world. As seen in section 4.4.4 there was originally a plan to include image tracking to add spacial alignment. If this were implemented, the application would have a slightly lower degree of artificiality.

7.8.5 Focus

My application has a focus on workspace, as it includes many objects of interest for the users to interact with. There is also a focus on the person because the other collaborators are visible. Ideally, the focus on the person should be higher with a more extensive representation of the other users.

7.8.6 Scenario

The scenario of the application is shared workspace, hence the focus on workspace awareness throughout the application. It is a workspace scenario because there is a huge focus on collaborative tasks in a virtual environment. This is arguably the most important aspect of the application.

7.9 Contributions

The first part of this section is a comparison between my application, and the second part is a list of my contributions.

7.9.1 Comparisons to Related Works

My application compares most to Ares, as it has a limited 3D representation of the user, users can manipulate existing 3D objects, write in 3D space, spawn and delete items. In addition to that, what users cannot do in Ares which they can do in my application is duplicating and resetting objects, as well as have a visible 3D representation of themselves. Another difference from Ares is the focus on a 3D controller, as well as a gizmo, which none of the related works had.

The 3D representation of users is less extensive than that of other existing works such as Spatiate, Spatial, or ScultprVR. The 3D representation is more of a stylized visualization of their controller than the users themselves. In this sense, it is closer to Spatiate where users can only see each other's controller and headset positions.

The way tools work is most similar to Tilt Brush and Spatiate, where users can create 3D geometry in the form of ribbon lines following the controller's position and rotation. The extent to which users can create varied geometry with different colors and effects, however, is very small in comparison to both these programs.

Features which are in my application, and not in any of the applications in related work and in few other AR applications, are the resource shelf which allows users to pull out 3D models, and the Gizmo which allows for users to perform much more precise and isolated transformations.

The biggest difference between my application and the related work is the focus on precise transformations which is not as present in the other applications. The number of isolated transformations possible with the Gizmo Tool is much higher than can be done with any of the other applications on HMDs. There are alternatives to the Gizmo in the research projects, but these are not run on an HMD, they are run on an iPad.

In figures 7.9.1 and 7.9.1 we can see a comparison of my end product referred to as "End Product" and the other applications in chapter 3. As we can see, not everything from the original tables was implemented in the end, missing are voice chat, and some of the elements from the 3D avatar.

Application	**VR**	**AR**	**Multiplayer**	**Controller**	**Gestures**
End Product		x	x	x	
Spatiate		x	x	x	
Spatial		x	x		x
SculptrVR	x		x	x	
Tilt Brush	x			x	
Ares		x	x		x
Immersive Group-to-Group Telepresence		x	x	x	x
Batmen Beyond		x		x	
SculptAR		x			x
AACT		x			x

Table 3: Table of feature comparisons between applications

7.9.2 List of Contributions

1. An application for cooperative 3D manipulation in AR on the MagicLeap One
2. A UI for precise manipulation with the Gizmo which can be tailored for different fields of use
3. Insight into what the average user thinks about the Gizmo as an AR tool
4. Insight into how to develop an intuitive AR interface for 3D manipulation
5. Insight into the strengths and weaknesses of development on the MagicLeap One
6. Insight into the strengths and weaknesses of the MagicLeap One as a cooperative tool in different fields
7. Insight into what developers and AR experts think about the MagicLeap One

7.10 Limitations

In this section I detail the multiple limitations which affected or might have affected my process and results negatively.

7.10.1 Inexperience With Unity and MagicLeap

I had some experience with Unity, but not much other than small hobby projects. I had also never used a MagicLeap One before and were unaware of the difficulties connected to it. This lead to many hours of experimentation with both Unity and the MagicLeap One. Many decisions and compromises were likely made due to lack of knowledge in the area.

7.10.2 Size of Project

Due to there being large libraries of resources available for AR HMDs like the Hololens, and VR in general, it was assumed that there would be many available resources for the MagicLeap One. This might have helped make a project this size more possible since fewer aspects would have to be programmed from scratch. But due to the lack of resources, this project became much larger than what might be feasible for one person in the span of a year. It is more realistic for this project to be iterated over several years.

7.10.3 Lack of Resources

As mentioned in section 7.1 there was a great lack of helpful resources for development. However, There was also a great lack of open-source and free 3D models which were relevant for the application. While doing user testing with different groups of users from different fields, I needed to gather 3D models which were relevant to them in order to connect better with the users. Time was spent searching for relevant 3D models which sometimes ended with making my own 3D models instead. This is time which could have been spent on development, and due to the number of different target groups would end up being a significant amount of time. Performing the number of user tests I did under development could therefore be argued as a double-edged sword, guiding us towards higher usability while also eating up development time.

7.10.4 Sample Size

Despite doing a fair number of user tests during development, the number of user tests I was able to perform for the final version were very few. Initial plans involved having tens of students visit the lab for testing and questionnaires. These were canceled because of Covid-19. Instead, I would have to make do with whomever I could manage to communicate with who were in possession of an MagicLeap One. This could have biased the result data, as four of the participants were developing applications on the MagicLeap One themselves. They were already familiar with the controller for example, so I was unable to test how difficult the application was to use for inexperienced users to the extent I wanted.

7.10.5 Focus on User Interface

In questionnaires, user tests and interviews, there was too much focus on the interface. This was aimed at gathering more information on how to make the application more user-friendly and increase the usability, but in the end made it more difficult to answer more important questions about the potentials of the application to strengthen learning and communication through collaboration. More focus should have been put on the collaborative parts of the application and how to strengthen it. This might have resulted in more focus on work space awareness during development.

7.10.6 Inexperience With Interviews

I have little experience with interviews and had close to no guidance on how to plan and conduct interviews. This resulted in us losing control over experts in interviews and not being able to get them to answer all the questions I wanted. On the other side I had problems with phrasing the questions for the user test participants in ways such that I received more fulfilling answers.

7.10.7 Video Demo Instead of User Test

As we can see in chapter 6, I was only able to get six users to perform an experiment with the application for user tests. The rest of the participants were only able to watch a video of the application.

These results might have been affected by the quality of the video, my performance in the video, or something I missed or focused too much on in the video. Because the participants were not able to try the application, they were not able to answer system usability questions either.

7.10.8 Rescheduling and Refocusing

As mentioned several times during the report, the application had to be refocused halfway through development. This was firstly because of research during user testing leading to new prioritization, but is without doubt also due to Covid-19 impacting the importance of features unpredictably. This affected development greatly, in having to rush decisions and development, having to cut features and bug fixes.

Application	Spatial Alignment	3D Avatar	Voice Chat	3D Sculpting	3D manipulation	3D drawing	Gizmo	Operations
End Product		Hand	x		x	x	x	Spawn object, delete, reset object, duplicate
Spatiate		Head	x		x	x		Spawn object, delete
Spatial		Torso, head, arms, hands	x		x			Spawn object, delete
SculptrVR		Torso, head, hands	x	x	x	x		Spawn object, delete
Tilt Brush			x	x	x	x		Spawn object, delete
Ares	x		x		x			Spawn object, delete
Immersive Group-to-Group Telepresence		Life Sized Reconstruction	x		x			
Batmen Beyond					x			Spawn object, delete, duplicate, undo, redo
SculptAR					x	x		Spawn object, delete, duplicate, undo, redo
AACT					x			Spawn object, undo, redo

Table 4: Table of feature comparisons between applications

8 Conclusion

My end goal in this thesis was to conduct research on the potential for collaborative work on the MagicLeap One in the educational setting, as well as explore other settings. my means of doing this was an application which I was going to develop for the MagicLeap One.

I performed literary research and studied existing 3D collaborative applications in AR and VR. There were two iterations of which I had multiple rounds of user testing. Once the Covid-19 lockdown started, testing and development were slowed down significantly.

I found answers to the research questions by creating an application which could be tested, and used it to conduct interviews and make questionnaires specifically tailored to answer the research questions. There were issues with the availability of test subjects and experts, but despite this I was able to get six different users to try the application and three different field experts. Through questionnaires, I was also able to gather opinions and suggestions from 24 people. Hopefully, this was enough data to draw a real conclusion.

The application was enough to conduct research on the potential of cooperation on the MagicLeap One to some extent. However in order to successfully research cooperation and especially educational content on the MagicLeap One the application needs to be developed further. The application runs in multiplayer, and serves as a working 3D cooperative manipulation application, but I believe the next step would be to tailor it for specific fields in order to do more meaningful research. It may have been more constructive to create an application specifically tailored for education from the start, however I believe doing so would have bloated the project and it would have been too difficult to develop in a year. This especially given the difficulties of developing the base application such as my end product. We can see through current research and research trends that AR is important for collaboration and learning, and it is important we conduct more meaningful research on this.

I believe choosing the MagicLeap One was the right choice given the available technology at the time. In general gesture recognition on AR HMDs was not at the level needed for precise 3D manipulation. The controller on the MagicLeap One made the choice worthwhile despite the difficulties that come with the platform, such as the lack of developer resources and community.

An assumption I made was that users would like the Gizmo to be as similar to their desktop application counterpart. After multiple rounds of testing however, it seems that the Gizmo itself is not something which can be easily transferred to AR, and needs to be tailored quite a bit to be intuitive. It is clear however due to the amount participants utilized the Gizmo in user tests, and their positivity for it in feedback despite difficulties with it, that it is definitely a useful addition to 3D manipulation.

I think it is important to acknowledge the fact that during the Covid-19 lockdown, I was not able to go back home because I was not able to move the MagicLeap One away from Trondheim. This really sheds light on the unavailability of the MagicLeap One. Ideally, I should have been able to go home and order a new HMD rather than move it, but this is not possible and indicates that the MagicLeap One is not available enough for use as of now.

Collaborative AR clearly has much unexplored potential still, and so does the technology we use for it. These AR HMDs are being developed at a fairly rapid pace. During the writing of this thesis, the Hololens 2 was released [66]. Apple is also working on their own AR HMDs [67]. More AR HMDs on the market could potentially increase enthusiasm and awareness of AR. As more of them become economically successful and available to the public, we can expect to see more research done on the topic and hopefully this will lead to more AR being used in education as well.

8.1 Future Work

There is still much to do in order to make an AR application which can successfully demonstrate the full capability the MagicLeap One has for cooperative 3D modeling and manipulation.

8.1.1 Add Feedback

Since users reported it was sometimes difficult to tell if something is happening or if they had selected an object successfully, it is clear there is still room for improvement with feedback.

There are multiple ways in which the feedback of the application could be improved. Currently, the feedback in the application is purely visual. The MagicLeap One controller can vibrate, and has LED lights around the touchpad. Additionally, sounds could be added for an extra layer of feedback. This could signify to the user that they are doing something correctly

For the Gizmo tool, vibration could be added when the user switches transformation mode with the touchpad. There could also be different sounds for each axis or transformation mode. When an object is transformed with the Gizmo, there could be a threshold which causes the controller to vibrate the controller in a pattern which seems like it is "clicking".

When grabbing or letting go of an object, there could also be a "click" sound accompanied by a little vibration to signify to the user that something happened.

There could also be sounds or vibrations when a user hovers over a new button on the context menu, and when a button is pushed. Other additional sounds could be a "pop" sound could be added when a user pulls something out of the resource shelf. Perhaps brush sounds could play when drawing. Sounds could be important because they not only indicate to the user that something is happening, but to their collaborators.

8.1.2 Add more UI elements

Users mentioned multiple times that some features seemed to be "hidden" since there were no UI indications that they were there.

When drawing for example, it was not visible to everyone where the actual line would land. Perhaps the tip of the cursor could be the same color as the color of the 3D drawing to indicate it as the "tip of the pencil".

During interviews and testing, some users reported it being difficult to predict exactly which item would be selected. A UI element which previews what would be selected could be useful here. For example, a marker which displays the point of intersection between the cursor or selection ray and the object it is intersecting with. There could also be a highlight which is visually distinct from the highlight of the selected object.

When the Grab tool is active, there is no indicator that an object can be scaled with the touchpad. An easy fix for this is to add UI elements constrained to the touchpad. For example a plus and a minus, which increase in size and make a noise or vibrate when pushed. Perhaps their size and amount of sound or vibration could depend on how far away from the center they are touching, as the amount of scaling depends on the distance from the center.

8.1.3 Improve 3D Avatar

As I learned in the expert interviews, an extensive 3D avatar is important for workspace awareness. The 3D avatar in my application is enough for a minimum viable product but not ideal.

The most important feature to add to the 3D avatar is a head model which follows the HMD position and rotation to convey gaze. Additionally, there could be a ray cast from the eyes to make gaze more obvious. Below the head there can additionally be a shape to indicate their torso position and increase their presence.

Above the head model there could be a name tag, or the 3D models themselves could be different colors, or both. This would help users differentiate from each other and tell which user is which.

If voice chat is added, there could also be a visual indicator on the 3D avatar that the user is speaking. There could for example be movement on the mouth if there is a mouth, or there could be a sound icon above their head.

Because of the low field of view on AR HMDs, no matter if a future version is written on the MagicLeap HMD or not, there could be visual indicators for other users where they are. This could for example be an arrow on their screen pointing to their direction.

The representation of the user's cursor could also be improved for other users, Currently, there is no indicator of which tool other users are using. Their tools should ideally change not only for themselves, but for the other users, so they can tell which tool they have active. Additionally, when a user activates a selection beam, it should be visible to other users. This could increase workspace awareness significantly as it becomes easier to point and for other users to see which objects they are selecting.

Speaking of selection, there can also be a visualization of which object other players have selected.

8.1.4 Save and Load

Currently, anything a user makes in the application will be deleted and reset when the application closes. This is because there is no saving or loading features. These should be added, because it would increase the usefulness of the application greatly. This could make it so users could make something one day, open it the next and continue work on it.

8.1.5 Improve the Context Menu

Currently, the context menu is at best usable. Feedback from questionnaires was that it was ugly, and feedback from experts was that it was too traditional. Perhaps rethinking the context menu completely would be beneficial.

Interviewees suggested for example making the context menu radial, or a set of 3D models which could be grabbed.

8.1.6 Persistent Content

The MagicLeap One is can recognize the environment it is in thanks to the continuous 3D scanning it does. This also means it can place content at a specific spot in space and having it stay there when the user leaves and returns later. This could be a useful feature for businesses. 3D content could for example persistently stay at specific desks in an office setting, or in different parts of a building in a construction setting.

Spacial Alignment

Spacial alignment has already been discussed a fair bit in 4.4.4. It is an essential feature for workspace awareness in colocated settings and should be implemented.

8.1.7 Improve Draw Tool

Currently, the Draw tool is very limited as it only draws in one size and color. Brush strokes cannot be selected, moved or deleted once they have been placed.

One immediate improvement on the Draw tool would be to allow brush strokes to be assigned the "selectable object" type the other 3D models are assigned in the application. This way they would inherit the select, transform, delete, reset, and duplicate functions the others have.

Additionally, there could be an "eraser" mode on the draw tool which allows the user to paint where they want to remove draw strokes. This could be a tool which removes only the polygons the eraser touches. This way deleting whole brush strokes could be reserved to the selection tool.

Another feature which was requested frequently is a color picker for the draw tool. This could be a UI element which pops up when the tool is active, or something the user can select in the context menu.

Finally, there could be a slider which would increase or decrease the brush stroke size. This could potentially be part of the color picker menu.

8.1.8 Improve Selection

The current selection system was rated in both interviews and questionnaires to have poor visibility. It is also fairly limited in that users can only select one object at a time.

Firstly the selection needs to be made clearer. An issue with selection currently is that it is not screen space, causing it to be less visible when models are scaled down. Writing a different highlighting shader which is based in screen space could fix this. Or there could be code which scales the thickness of the highlight up when objects are scaled down.

One feature which was requested multiple times is multi object selection. This could open possibilities for more sophisticated features such as moving multiple objects at once and other operations described in 8.1.9. Multi object selection could be implemented with an alternative selection mode in which users can define the bounding box of selection by dragging the cursor, similarly to how box selection on 2D screens work. Alternatively, there could be some button which activates multi object selection in that it adds every selected object to a collection of selections until the user turns it off again. An issue with this is that it could be annoying for users to have to swap between selection modes, so there would have to be more research done on the subject.

Selection should also be visible to other users, because this will increase workspace awareness by letting other users know exactly which objects are being worked on.

8.1.9 More Sophisticated Object Interactions

Multiple users expressed an interest in putting together objects to permanently create a new object. The ability to select a series of objects and convert them to a single object which could be duplicated would increase the amount of modeling possibilities greatly.

In addition, there is a need for adjusting the resource shelf. An expert suggested being able to drag an object into the resource shelf to replace the existing object with the new object. This could be way users would create new objects as well.

There also needs to be some form of import and export of 3D models. This means there needs to be additional code for handling the processing of 3D models for the application, turning from a 3D model file to a selectable object in the application.

8.1.10 Bug Fixes

Finally, there are many bugs which need fixing in the application. Most likely many are not discovered yet, but the list includes:

- Multiple objects can be selected while the context menu is open, which can crash the application
- The context menu cannot be navigated with the touchpad in multiplayer
- A user quitting the application will freeze it for everyone else
- A user going to the MagicLeap main menu while in the application freezes it for everyone else
- Multiple users can select an object at the same time which causes conflicts
- The head on the 3D avatar disappears when the application starts
- If a user duplicates an object, another user cannot duplicate that specific duplicated object
- Objects in the resource shelf can be deleted
- Photon does not send enough data for drawing to work correctly in multiplayer, parts of the lines will be missing which causes several errors per second for other players and lag

8.1.11 Final Requirements

As a suggestion, based on the results and discussions, I made a final version of the requirements for the application.

Functional Requirements

1. The UI needs to mitigate the effects of the low field of view and unfamiliarity of controller
2. Manipulate 3D objects
3. Spatial Alignment:
   The objects in the virtual space need to be in the same location for every user in the same room. This will most likely remove a great deal of confusion for colocated users.
4. Multiplayer
5. Add and delete 3D objects
6. Select objects from up close and afar
7. Gesture and point at objects
8. Hide and un-hide objects
9. Select objects from a list
10. Selection should be visible for every user
    In order to increase workspace awareness, every user should be able to see which objects are selected, and ideally by whom.
11. Draw in 3D space:
    Drawing as an annotation tool adds a lot to communication. I noticed in user tests this feature was not only fun to use but also described in feedback as something which could be very useful if developed further.
12. Manipulate 3D drawings:
    Drawings need to be registered as objects similar to everything else, it needs to be possible to manipulate, reset, delete, and duplicate 3D drawing objects.
13. Voice communication:
    Voice communication has to be present in order to maximize communication.
14. Save and load:
    In order to work on objects over multiple sessions there needs to be functionality to save and load a scene.
15. Export and import:
    In order to get 3D objects to and from other applications, exporting and importing of 3D objects needs to be in place.
16.

Non-Functional Requirements

1. Context menu needs to be easily maneuverable:
   Since there are so few buttons on the controller, there needs to be a simple system put in to navigate the UI. This is also an important part of UI in general, and not just this application.
2. Operations need to be simple to perform
3. The UI should avoid relying on memory
4. Someone who has never used the application before should be able to use it themselves
5. Holograms must stay in place with minimal drift unless moved by the user

6. The hardware must be available for purchase:
   Due to my inability to go home while developing for the MagicLeap One, I would like to stress that the hardware the application runs on must be available for purchase and more easily accessible. This requirement is replacing the requirement of the application running on the MagicLeap One.
7. The operating system of the hardware must be continuously updated so that it is compatible with the latest development platforms
8. The Gizmo needs to be nonthreatening to new users

9 Additional Figures

9.1 Other HMDs

Figure 79: The Vive HMD and motion controllers, as well as the IR cameras.

Figure 80: From left to right: Oculus Rift and Oculus Go HMDs and their respective motion controllers.

Figure 81: The Google Glass HMD

10 Additional results figures

10.1 Questionnaire During Second Iteration

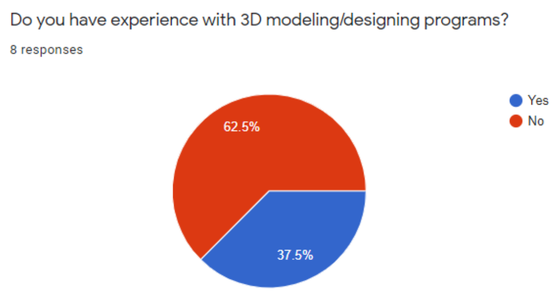

Figure 82: Question 1

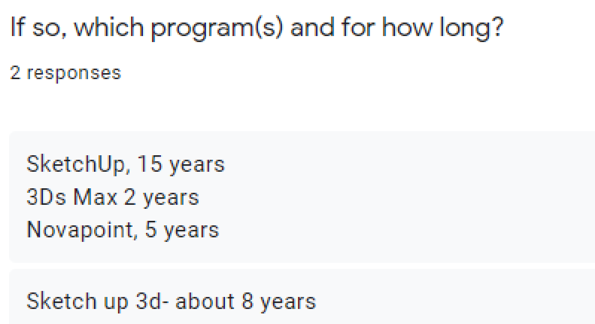

Figure 83: Question 2

10.2 Final Results Tables

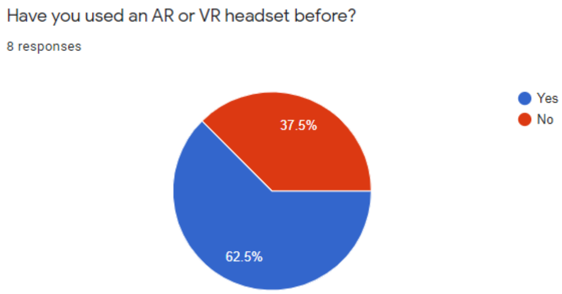

Figure 84: Question 3

Figure 85: Question 4

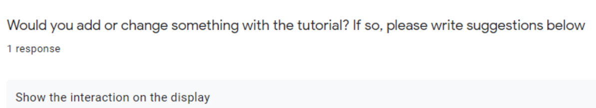

Figure 86: Question 6

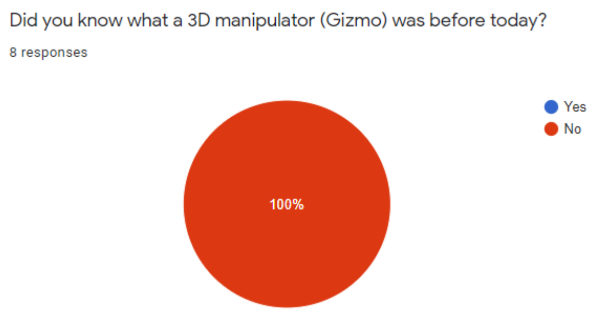

Figure 87: Question 8

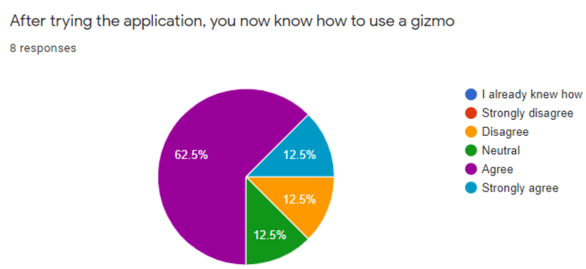

Figure 88: Question 9

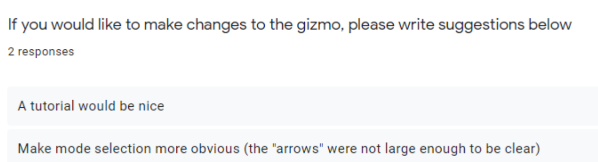

Figure 89: Question 16

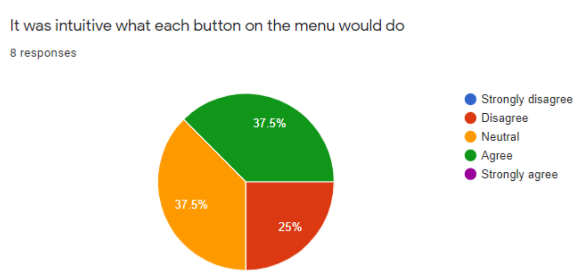

Figure 90: Question 23

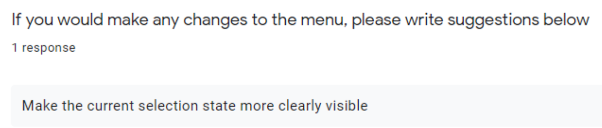

Figure 91: Question 24

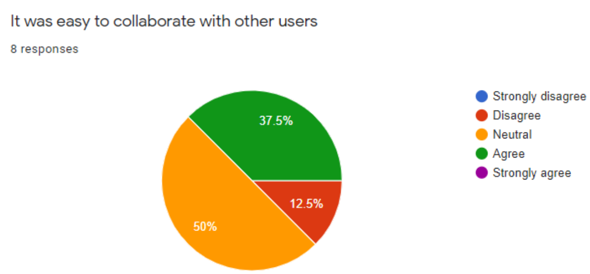

Figure 92: Question 25

ID	Response
1	Blender
2	Blender
3	Oculus Medium
4	Blender
5	Blender
6	Blender
7	Maya, Blender
8	Maya, Sketchup
9	Sketch 3D

Table 5: Question 4: Do you have any experience with 3D software, if so, which?

ID	Response
1	No change atm.
2	Make it more visible that the object was selected, maybe the box the object is in could change color
3	Change the outline color on selected objects, or make it configurable if possible.
4	Higher Colour Contrast around the objects
5	I would add a bolder outline when an object is selected. Alternatively I would use a colour instead of white around model. It was a bit difficult to see when the oil platform was selected as both the model and the outline were white and the small details on the model blended somewhat with the thin outline.
6	Stronger signal of selection
7	Maybe have a more distinct colour outline around the object, or a circle or square (or a gradient version of these) in the background of the object, or have a more clear feedback when you select an object (maybe a small click sound).
8	I realized only upon rewatching that the object actually has a visual "highlight" to it when selected, because you pointed it out. I didn't really see it all that well, so I would like to either have a lot more contrast to the highlight, or some other type of visual cue alltogether.
9	(I assume you mean selecting an item in the shelf at around 1:31?) You mentioned an outline, but I struggled to see it in the video, so you'd maybe want something more visually aparent. I would maybe have the outline be thicker, or maybe recolor the containing shelf box whenever you select something.
10	outline in another color
11	I would make sure the color of the highlight differed from the color of the 3D model.
12	A larger outline would be nice
13	Maybe a more visible way to highlight a selected object, for example the color of the box border around the object changes when selected
14	better highlight?
15	I wouldn't change it
16	Giving a different or more visible highlight color
17	More pronounced highlighting, perhaps by using a stronger color or thicker border
18	Nop
19	Might be because I am not familiar with the Leap, but holding the bumper seems unnatural. Using the trigger would be my first instinct. Having a clearer outline or a shading to convey that an object is chosen would have helped as well.
20	Maybe even stronger color / border
21	The ray seems fine and probably more useful in a busy environment. The main issue js the outline on the selected object needs more stroke

Table 6: Question 12: How would you change the way objects are selected?

ID	Feedback
1	Maybe the tool selected could be written somewhere, when you are not inside the menu
2	Change the color of the button for the active tool.
3	Round the corners of the menu and make a slightly transparent background for it. layout is otherwise very good
4	I would add a bit more padding around the menu titles
5	No opinion
6	It was a little difficult to see that a button was selected, I didn't see the "title" at first. Maybe make this a bit different than the rest of the text. Maybe add a "Selected:" text.
7	Some icons in addition to the text labels would make it even quicker to scan through the options, and understand what their function is.
8	I would like it more aesthetically pleasing.
9	Highlight the active mode's button when re-entering the context menu
10	Make it a little bit bigger
11	Wouldnt
12	Marking the button of the selected tool instead of (or in addition to) having it written on top. Having an extra step when deleting to prevent accidental deletes, and/or having an undo button
13	Bigger buttons. Maybe also have to option to just select using a dot on the center of the screen (however, it may be confusing if you want the other options instead)
14	Make them selectable with the touchpad as well as cursor/ray
15	Would love some bolder font and icons.

Table 7: Question 15: How would you change the context menu?

ID	Feedback
1	Maybe write the selected transform mode somewhere when you are not inside the menu
2	an option to let the user see the parameters of the current transformation (ie a window showing: move (x: 10.0, y: 0.0, z: 4.0))
3	tutorial pop-up instructions
4	Not really
5	It is very easy to notice something is different in rotate mode, but I t.wonder if the difference between the similarly sized arrows (move) and boxes (scale) is too slight, and that some people might not notice i
6	short animation of a translation / rotation / scale along the axis?
7	Hmm nop
8	Really liked the Gizmo. Only thing preventing the "Strongly Agrees" were that seeing what was selected would be nice to have n the Gizmo-wheel itself. One might not know right away that a blue line with cubes means scale Z.
9	Having some way of making the user understand that the round mini menu can only select one option on each half circle. Maybe use some radio button vizualization such that the user see the constraints and does not have to try it out.
10	Not really. Pretty self explainatory

Table 8: Question 19: Can you think of any ways to make it easier to understand what the gizmo does?

ID	Feedback
1	Wouldn't change it
2	Perhaps let the user toggle the axes instead of selecting all or one.
3	Maybe the cursor can change (either color or shape), when the tool is "grabbing"?
4	Maybe highlight where you grab the object, so that if you grab inside the model you can still see the point you will rotate the model around.
5	grabbing animation + sound, holding animation + sound?
6	Wouldn't
7	No, it is perfect
8	N/A

Table 9: Question 21: Can you think of any ways to make it easier to understand what the grab tool does?

ID	Feedback
1	Wouldn't change it
2	let the user select the color they are drawing with
3	change the selector arrow to a pencil that shows the angle of the arrow a bit better
4	It seems you need to be careful not to write too small, in order for your text to be legible. Maybe it would have been nice to have a few pre-made phrases to select from, like "Good!", "Needs to be changed", etc. It can also be difficult to read if you don't see it from the correct angle. Would it be possible for text to be a 2D-sprite-type-thing that always faces the player?
5	In the video it was hard to tell if its easy to percieve the depth of the drawing.
6	Passer kanskje best til grove tegninger, vanskelig med detaljer i tegning?
7	Smoother and more responsive
8	Could be that smooth lines could be nice, but you probably know that already. An idea could be to have some way to separate texts / drawings as it could be hard to distinguish each of them if they are close. Maybe add come background. Could be just making a virtual paper and make it possible to draw on that and move/scale/rotate that around.
9	An option to grab and resize text

Table 10: Question 23: What changes would you make to convey what the other players are doing better?

ID	Feedback
1	Make it more visible where the other player is, maybe a name beside their cursor
2	Something visually to tie the other players tool to the object they are manipulating, could be just a line.
3	put in an indicator for where their head is so you can see what they're looking at
4	Maybe have the other player's tools different from your own (maybe it's easier to understand when you're doing it yourself, though).
5	I was a bit confused that the other players gizmo seemed to be in "rotate mode", when he was just moving and grabbing. I would like to be able to see what modes other players are in, and what objects they have selected.
6	Didnt notice if this was a feature, but it'd be nice to know what object the other player currently has selected. It also seems somewhat difficult to know where the other player is (specifically when they're outside your field of view, or behind the model you're making), so Id might make their pointer be visible throught objects with an outline, and maybe have a small arrow showing you which direction they are in when they are outside your FoV
7	Maybe change the highlight of an object the other person has selected so that its different.
8	Show their goggle and hand positions
9	Maybe make the other player more "visible" change the color of their pointer for example
10	talking animation? some simple user icon or model?
11	There is no way to see who are interacting with an object when the ray is used
12	As long as the pointer is visible, I would be happy. More info from the user is great too, but not important to me.
13	Colored cursors

Table 11: Question 26: What changes would you make to make cooperation between players better?

ID	Feedback
1	A nametag above each players tool.
2	I can imagine with more than two players it can get confusing, so I would definitely add some nametags, or other forms of identification.
3	To be honest, there was not a lot of cooperation shown in the video, but if it all works smoothly then I can see this being pretty good for collaboration
4	se each others pointers
5	Better physical presence like the answer to Q25
6	Maybe a chat, but that may be hard in VR, so speaking is sufficient enough. Maybe adding notes on paper (previous idea) could be nice.

Table 12: Question 28: What changes would you make to make cooperation between players better?

ID	Feedback
1	annotations that show up when an object is selected.
2	Momentum of objects by throwing them or interacting with them by the users
3	Customisation - upload own models, select self defined amount of inventory slots
4	Maybe if you could include "suggestions" of changes that students can propose, maybe add highlights, "post its".
5	Maybe some way to easily pin objects to a surface, like a desk, seeing as you usually sit down in that context?
6	If you could design interactive models where some of the models parameters were locked. Then a teacher could create a model of for example a map of battle in history and the students could advance "time" which would then change the positon of objects in a preplanned way the teacher had set up.
7	It would have to be extended to fit the subjects context, but I can see it being very useful.
8	Visualisation is a power tool that can be used to show complex theories that may be come simpler to just show. Cooperation across long distances is also a great benefit.
9	import / export of models?
10	Notater
11	Yes
12	Make it possible to animate based on time. That could open up a whole new world. Making it possible to simulate phenomenons could also create more understanding. The more you could create, the more learners are able to be their own creators of their own intellectual structures.
13	A spectator option that forbids interaction with objects

Table 13: Question 34: Can you imagine a change that would make it more useful for education?

ID	Feedback
1	Options to change the materials on objects.
2	being able to sculpt objects by pressing a key to make the objects go into negative space while the arrow approaches
3	Different types of brushes, brush sizes and colours in drawing tool
4	Addition of more "MS Paint"-like tools
5	When the art is made: Maybe when you select an object (sculpture, painting etc), information about it appears in a way (sound, text, video). When making art: being able to understand how different materials act with each other.
6	Undo/redo button
7	I'd add more focus on sculpting models, as opposed to just placing figures.
8	More drawing tools
9	A ctrl+z function if one is not present.
10	More colors for the drawing tool, more brushes, ways to edit models.
11	Visualisation and the possibility of cooperation on an artistic digital project. 3D modelling, for example, with the tools presented (and maybe add more complex tools) would be great for tutorials and such.
12	More tools. Making it easy to add objects, save objects and selecting transforming generic building pieces for more creative freedom
13	Make it possible to add more filters, maybe also make everything changeable. Like make it possible to drag all vertices freely in any model. Maybe warp all shapes with some tools. Maybe be able to color some parts of a model, too.
14	Shape tools

Table 14: Question 36: Can you imagine a change that would make it more useful for artists?

ID	Feedback
1	A grid + snapping options. showing the coordinates of objects
2	momentum of objects
3	AR without the use of a vr gear or anyhow a less big chunky thing on your head.
4	If you could look at the same space with others and see where they are in their space it could assist in holding long distance meetings, like during lockdown.
5	It really depends on the business context.
6	Businesses often operate with complex theories and structures that can be simplified through visualisation. This can further make it easier to communicate with business partners/customers. Again, location would not be an issue as well. Any room or place could become your office. This would also be more immserive and interactive than just using voice chat or text format.
7	No

Table 15: Question 39: Can you imagine a change that would make it more useful for business?

ID	Feedback
1	astronomy
2	Psychology
3	Museums, and maybe some cool art-installations or concert opportunities.
4	Sales
5	3D-design in general
6	No
7	Entertainment. Allows live practice staging or brainstorming

Table 16: Question 40: Are there any other fields you think it could be useful?

Bibliography

[1] Peterson, D. & Robertson, C. Utilizing virtual reality in teaching and training: Progress toward virtual surgical simulations Higher-order brain circuits View project UTILIZING VIRTUAL REALITY IN TEACHING AND TRAINING: PROGRESS TOWARD VIRTUAL SURGICAL SIMULATIONS. Technical report, 2014. URL: https://www.researchgate.net/publication/235800900.

[2] Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. 12 1995. <title>Augmented reality: a class of displays on the reality-virtuality continuum</title>. In *Telemanipulator and Telepresence Technologies*, volume 2351, 282–292. SPIE. doi:10.1117/12.197321.

[3] What is Augmented Reality? | Interaction Design Foundation. URL: https://www.interaction-design.org/literature/topics/augmented-reality.

[4] 'Pokemon Go' Creator on Augmented Reality's Massive Potential – Variety. URL: https://variety.com/2019/gaming/features/pokemon-go-creator-on-augmented-realitys-massive-potential-1203169992/.

[5] Glass – Glass. URL: https://www.google.com/glass/start/.

[6] How Augmented Reality Enhances Manufacturing Factories - GE Reports. URL: https://www.ge.com/reports/smart-specs-ok-glass-fix-jet-engine/.

[7] Virtual Reality vs. Augmented Reality vs. Mixed Reality - Intel. URL: https://www.intel.com/content/www/us/en/tech-tips-and-tricks/virtual-reality-vs-augmented-reality.html.

[8] Virtual reality headsets given to terminally ill patients so they can enter computer simulations during their final days. URL: https://www.thesun.co.uk/tech/3464454/virtual-reality-headsets-given-to-terminally-ill-patients-so-they-can-enter-computer-simulatio

[9] VR Gives Terminally Ill Children the Experience of a Lifetime | OZY. URL: https://www.ozy.com/good-sht/vr-gives-terminally-ill-children-the-experience-of-a-lifetime/94159/.

[10] Wang, X. & Dunston, P. S. GROUPWARE CONCEPTS FOR AUGMENTED REALITY MEDIATED HUMAN-TO-HUMAN COLLABORATION. Technical report, 2006.

[11] Ens, B., Lanir, J., Tang, A., Bateman, S., Lee, G., Piumsomboon, T., & Billinghurst, M. 11 2019. Revisiting collaboration through mixed reality: The evolution of groupware. *International Journal of Human Computer Studies*, 131, 81–98. doi:10.1016/j.ijhcs.2019.05.011.

[12] Gutwin, C. & Greenberg, S. 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work*, 11(3-4), 411–446. doi:10.1023/A:1021271517844.

[13] Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., & Mitsuishi, M. 2000. GestureMan. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, 155–162, New York, New York, USA. Association for Computing Machinery (ACM). URL: http://portal.acm.org/citation.cfm?doid=358916.358986, doi:10.1145/358916.358986.

[14] Kim, K., Billinghurst, M., Bruder, G., Duh, H. B. L., & Welch, G. F. 11 2018. Revisiting trends in augmented reality research: A review of the 2nd Decade of ISMAR (2008-2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11), 2947–2962. doi:10.1109/TVCG.2018.2868591.

[15] Zhou, F., Dun, H. B. L., & Billinghurst, M. 2008. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008*, 193–202. doi:10.1109/ISMAR.2008.4637362.

[16] Wittich, W., Lorenzini, M. C., Markowitz, S. N., Tolentino, M., Gartner, S. A., Goldstein, J. E., & Dagnelie, G. 9 2018. The Effect of a Head-mounted Low Vision Device on Visual Function. *Optometry and Vision Science*, 95(9), 774–784. doi:10.1097/OPX.0000000000001262.

[17] HP VR Backpack PC | HP® Official Site. URL: https://www8.hp.com/us/en/vr/vr-backpack.html.

[18] Robinett, W. & Rolland, J. P. 1 1992. A Computational Model for the Stereoscopic Optics of a Head-Mounted Display. *Presence: Teleoperators and Virtual Environments*, 1(1), 45–62. doi:10.1162/pres.1992.1.1.45.

[19] Hu, L. & Bi, X. 2013. Application research of virtual reality technology in electronic technique teaching. In *Advances in Intelligent Systems and Computing*, volume 180 AISC, 153–159. Springer Verlag. doi:10.1007/978-3-642-31656-2{_}22.

[20] Developing for HoloLens | Build 2015 | Channel 9. URL: https://channel9.msdn.com/Events/Build/2015/C9-08.

[21] With HoloLens, Microsoft aims to avoid Google's mistakes - Reuters. URL: https://www.reuters.com/article/us-microsoft-hololens-idUSKCN0YE1LZ.

[22] FACT SHEET TABLE OF CONTENTS Magic Leap One Creator Edition Pages 2-3 Magic Leap One Creator Edition Product Specifications Pages 4-5 Magic Leap Corporate Page 1 How to Purchase Magic Leap One Creator Edition. Technical report.

[23] Magic Leap 1 | Magic Leap. URL: https://www.magicleap.com/en-us/magic-leap-1.

[24] Magic Leap CEO: Augmented Reality Could Replace Smartphones. URL: https://www.forbes.com/sites/ellenhuet/2015/02/24/magic-leap-ceo-augmented-reality-could-replace-smartphones/#68c1a2e42680.

[25] Magic Leap CEO Teases 'Golden Tickets' for Its Augmented-Reality Device | WIRED. URL: https://www.wired.com/2015/02/magic-leap-reddit/.

[26] Magic Leap One 6DoF Controller In-depth. URL: https://arcritic.com/2278/magic-leap-one-6dof-controller-in-depth/.

[27] (No Title). URL: https://www.ist.ucf.edu/Portals/10/files/RAPIDposters/RAPIDFOV_Oculus_Vive_Poster.pdf?ver=2019-06-04-095455-067.

[28] Viewing Frustum & Clipping Plane | Magic Leap. URL: https://developer.magicleap.com/en-us/learn/guides/field-of-view.

[29] Report: Magic Leap One's Field of View Detailed in Developer Documentation – Road to VR. URL: https://www.roadtovr.com/report-magic-leap-ones-field-view-detailed-developer-documentation/.

[30] HoloLens (1st gen) hardware | Microsoft Docs. URL: https://docs.microsoft.com/en-us/hololens/hololens1-hardware.

[31] Use the HoloLens clicker | Microsoft Docs. URL: https://docs.microsoft.com/en-us/hololens/hololens1-clicker.

[32] Pan, Z., Cheok, A. D., Yang, H., Zhu, J., & Shi, J. 2 2006. Virtual reality and mixed reality for virtual learning environments. *Computers and Graphics (Pergamon)*, 30(1), 20–28. doi: 10.1016/j.cag.2005.10.004.

[33] Radu, I. Augmented reality in education: a meta-review and cross-media analysis. doi: 10.1007/s00779-013-0747-y.

[34] Sommerauer, P. & Müller, O. Association for Information Systems AIS Electronic Library (AISeL) AUGMENTED REALITY FOR TEACHING AND LEARNING-A LITERATURE REVIEW ON THEORETICAL AND EMPIRICAL FOUNDATIONS. Technical report. URL: https://aisel.aisnet.org/ecis2018_rp/31.

[35] Unity - Manual: Multiplayer and Networking. URL: https://docs.unity3d.com/Manual/UNet.html.

[36] Photon PUN Pricing Plans | Photon Engine. URL: https://www.photonengine.com/en-US/PUN/pricing#plan-20.

[37] How to build a real-time multiplayer game using only Firebase as a backend | by Koya Tamura | Medium. URL: https://medium.com/@ktamura_74189/how-to-build-a-real-time-multiplayer-game-using-only-firebase-as-a-backend-b5bb805c6543.

[38] gizmo noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com. URL: https://www.oxfordlearnersdictionaries.com/definition/english/gizmo?q=gizmo.

[39] About Using 3D Gizmos | AutoCAD 2020 | Autodesk Knowledge Network. URL: https://knowledge.autodesk.com/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/AutoCAD-Core/files/GUID-7BD066C9-31BA-4D47-8064-2F9CF268FA15-htm.html.

[40] Companies Using Autodesk 3ds Max, Market Share, Customers and Competitors. URL: https://discovery.hgdata.com/product/autodesk-3ds-max.

[41] Use Center Flyout | 3ds Max 2018 | Autodesk Knowledge Network. URL: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/3DSMax-Basics/files/GUID-57814FDB-4D6B-469D-9D46-84380E12C8C3-htm.html.

[42] Companies Using Blender, Market Share, Customers and Competitors. URL: https://discovery.hgdata.com/product/blender.

[43] Japanese anime studio Khara moving to Blender — blender.org. URL: https://www.blender.org/user-stories/japanese-anime-studio-khara-moving-to-blender/.

[44] The Pixol | ZBrush Docs. URL: http://docs.pixologic.com/getting-started/basic-concepts/the-pixol/.

[45] ZTools and Edit mode | ZBrush Docs. URL: http://docs.pixologic.com/getting-started/basic-concepts/ztools-and-edit-mode/.

[46] ARES - Augmented Reality Engineering Space | XRGO Business Software. URL: https://xrgo.io/en/product/ares/.

[47] ARES (Augmented Reality Engineering Space): 3D CAD Visualization & Manipulation - YouTube. URL: https://www.youtube.com/watch?v=7fKyna9Q4GQ&t=3s.

[48] The latest Tilt Brush tool is a game-changer for VR artists | Engadget. URL: https://www.engadget.com/2018-07-13-tilt-brush-game-changer-vr-artists-3d.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=

AQAAAL-kJjXsrzU3xpr8iypJg7s0eR7bwYsgCUduC7jhmZHaQXZjhXl7UtllJMdbDsM3mGiD3kUTRClzpNHY61_7w5eCgWqO5Uw4jUsrD5O_dQEZ-nNVxcyRaRtcvX8Fnd3Mbajgh0X0BhtJDppWjxNq4_djV3KIHVvgslu4ozgMwiZO.

[49] Tilt Brush Tutorials: Selection Tool - YouTube. URL: https://www.youtube.com/watch?v=hD-uY6fJLeg.

[50] Beck, S., Kunert, A., Kulik, A., & Froehlich, B. 4 2013. Immersive group-to-group telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 19(4), 616–625. doi:10.1109/TVCG.2013.33.

[51] Buxton, W. A. S. Telepresence: Integrating Shared Task and Person Spaces. Technical report.

[52] Rodrigues, A. M., Belloc, O., Borba, E. Z., Nagamura, M., & Zuffo, M. K. 4 2017. Batmen beyond: Natural 3D manipulation with the BatWand. In *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, 258–259. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/3DUI.2017.7893370.

[53] Sangalli, V. A., De Oliveira, T. V., Soares, L. P., & Pinho, M. S. 4 2017. SculptAR: An augmented reality interaction system. In *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, 260–261. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/3DUI.2017.7893371.

[54] Bhargava, A., Bertrand, J., & Babu, S. V. 4 2017. AACT: A mobile Augmented Reality application for Art Creation. In *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, 254–255. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/3DUI.2017.7893368.

[55] Introduction | Photon Engine. URL: https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro.

[56] Developer Portal | Magic Leap. URL: https://developer.magicleap.com/en-us/learn/guides/spatialalignment-mltk.

[57] Unity VR Tutorial: How To Build Tilt Brush From Scratch - YouTube. URL: https://www.youtube.com/watch?v=eMJATZI0A7c&t=1764s.

[58] Getting started with MRTK | Mixed Reality Toolkit Documentation. URL: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/GettingStartedWithTheMRTK.html.

[59] MakeCode for Minecraft makes learning to code super fun. URL: https://www.microsoft.com/en-us/research/blog/code-minecraft/.

[60] Developer Portal | Magic Leap. URL: https://developer.magicleap.com/en-us/learn/guides/web-platform-overview.

[61] Zero Iteration | Magic Leap. URL: https://developer.magicleap.com/en-us/learn/guides/lab-zi.

[62] Charting a New Course | Magic Leap. URL: https://www.magicleap.com/en-us/news/news/charting-a-new-course.

[63] GitHub - thinkmobiles/ARInvestigation-ARToolKitTest. URL: https://github.com/thinkmobiles/ARInvestigation-ARToolKitTest.

[64] Magic Leap - Dissonance: Unity Voice Chat. URL: https://dissonance.readthedocs.io/en/latest/Platforms/Magic%20Leap/.

[65] The Ecological Approach to Visual Perception - James Jerome Gibson - Google Bøker. URL: https://books.google.no/books?redir_esc=y&hl=no&id=DrhCCWmJpWUC&q=affordance#v=snippet&q=what%20it%20offers%20the%20animal&f=false.

[66] HoloLens 2: Find Specs and Features - Microsoft HoloLens 2. URL: https://www.microsoft.com/en-us/p/holoLens-2/91pnzzznzwcp/?activetab=pivot%3aoverviewtab.

[67] Go read this detailed look at Apple's AR and VR headset development - The Verge. URL: https://www.theverge.com/2020/6/19/21296674/apple-vr-ar-headset-glasses-external-hub-jony-ive-bloomberg-go-read-this.