

University of Nevada, Reno

Virtual Reality: Locomotion, Multi-player, and External Camera Integration

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in Computer Science and Engineering

by

Andrew Thomas Flangas

Dr. Frederick C. Harris Jr., Advisor

May, 2021

© by Andrew Thomas Flangas
All Rights Reserved



THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Andrew T. Flangas

Entitled

Virtual Reality: Locomotion, Multi-player, and External Camera Integration

be accepted in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Dr. Frederick C. Harris, Jr., Advisor

Dr. Sergiu M. Dascalu, Committee Member

Dr. Javad Sattarvand, Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

May, 2021

Abstract

The field of virtual reality (VR) continues to expand and evolve with each passing year, and in doing so, continues to attract the interest of many. In this thesis we look at three areas. One area of research is a case study comparing two types of locomotion, a natural and inorganic method of movement. Aspects such as completion time and efficiency are measured, as well as personal preferences for the participants favorite form of locomotion. The second research area is the creation process of a multi-user cooperative VR game is described in detail. This game was created using the Unity gaming engine, along with the Blender modeling tool to create the 3D models. The third area focuses on two methods to merge live webcam feeds together into a single panoramic frame to be viewed in a VR environment and hopefully lead to control of a remote mining machine.

Dedication

I dedicate this thesis to my grandpa William G. Flangas for ingraining in my mind the importance of a college education, and my uncle Albert Flangas who taught me that life is too short to not have a good time.

Acknowledgments

I would like to thank my committee Dr. Harris, Dr. Dascalu, and Dr. Sattarvand for their invaluable guidance throughout my graduate studies. I would also like to thank my parents and grandma for aiding me in many ways throughout my college career and always being there for me when I needed them. Next I would like to thank my older brother William for taking the time to drive me across the country for my new job in Huntsville, Alabama. I am also truly grateful for the friends I made along the way in the Computer Science and Engineering department at the University of Nevada, Reno. I felt I was embraced by the members of this department, both students and faculty, despite coming from an Electrical Engineering background for my undergrad.

Contents

Abstract	i
Dedication	ii
Acknowledgments	iii
List of Figures	vii
1 Introduction	1
2 Locomotion in VR: A Comparison Between a Natural and an Inorganic Locomotion Technique	3
2.1 Introduction	4
2.2 Implementation	5
2.3 User Study Methodology	6
2.3.1 Participants	6
2.3.2 Apparatus	7
2.3.3 Procedure	8
2.3.4 Tasks	9
2.3.5 Design	10
2.4 Results	11
2.4.1 Course Data	11
2.4.2 Cybersickness Data	12
2.4.3 Fatigue Data	12
2.5 Discussion	13
2.5.1 Comfort	13
2.5.2 Intuitiveness	14
2.5.3 Immersion	14
2.5.4 Cybersickness	15
2.5.5 Fatigue	16
2.5.6 Efficiency	16
2.5.7 Enjoyment	18
2.6 Conclusions and Future Work	18
2.6.1 Conclusions	18
2.6.2 Future Work	19

2.7	Acknowledgements	20
3	Multi-player Game: Multi-User VR Cooperative Puzzle Game	21
3.1	Introduction	22
3.2	The Creation Process	22
3.2.1	Blender Modeling	22
3.2.2	Development in Unity	25
3.2.3	Developing the Powers	26
3.2.4	Level Design	31
3.2.5	Mirror Networking	33
3.3	Gameplay	35
3.3.1	Locomotion	35
3.3.2	Positive Outcomes	35
3.4	Conclusions and Future Work	36
3.4.1	Conclusions	36
3.4.2	Future Work	37
4	External Camera Integration: Merging Live Video Feeds for Remote Monitoring of a Mining Machine	38
4.1	Introduction	39
4.2	Concept	42
4.2.1	The Problem	43
4.2.2	Tools for Testing	43
4.3	Requirements	44
4.3.1	Stakeholder Interviews	44
4.3.2	Software Requirements	46
4.4	Use Case Modeling	47
4.5	Design and Implementation	49
4.5.1	Block Diagram	49
4.5.2	Design Class Diagram	49
4.5.3	Data Design	52
4.6	Merging Live Video Feeds and Testing Camera Placement	52
4.6.1	Using OpenCV for Python to Merge Webcam Footage	52
4.6.2	Loading Webcam Footage Directly to Unity	55
4.6.3	Testing Placement of Cameras on the Machine Using Unity	55
4.7	Related Work	57
4.7.1	VR and Robotics	57
4.7.2	Relevant Projects and Applications	59
4.7.3	Other Related Areas	60
4.8	Conclusion and Future Work	62
5	Conclusions and Future Work	65
5.1	Conclusions	65
5.2	Future Work	66

5.2.1	Testing Other Forms of Locomotion	66
5.2.2	Improving the Multi-user Cooperative VR Game	67
5.2.3	Merging Live-streams for Remote Cameras	67
5.2.4	Implementing a Joystick Controller to Mining Machine Project	68
References		69

List of Figures

2.1	A perspective view of what participants would see when the testing application starts	8
2.2	An overview of the course for each participant to follow.	10
2.3	Box and Whisker Plot for the course completion times of each participant in seconds. Inorganic: Average = 156.8, Median = 156.2, Outlier = 166.6, Maximum = 163.4 Minimum = 150.7 Natural: Average = 167.7, Median = 166.8, Maximum = 196.3, Minimum = 130.6	11
2.4	Box and Whisker Plot for Cybersickness Responses. Inorganic: Average = 1.55, Median = 1, Outlier = 4, Maximum = 3 Natural: Average = 2.3, Median = 1.5, Maximum = 5	12
2.5	Box and Whisker Plot for fatigue responses. Inorganic: Average = 1.15, Median = 1, Outlier = 2, Maximum = 1 Natural: Average = 3.55, Median = 3, Maximum = 8	13
3.1	A Screenshot in Blender which shows how to setup animations using the human rig for the witch model.	23
3.2	Crystal Balls representing powers on their Stands	24
3.3	The Octopus model	25
3.4	A first person view showing the object that the swap power applies to has its material changed to red. After the swap power is applied, the player will switch places with the cube.	26
3.5	The shrink power scales the player to a considerably smaller size. Notice the crystal ball stands next to the player.	27
3.6	A split screen (two images from different player's screens). When the ice power is selected and activated it changes the material of your witch's skin into a light blue color. The ice power is bringing down the balloon seen in the right window.	28
3.7	The fireball is used for the bomb explosion power.	29
3.8	A split screen (two images from different player's screens). When the fire power is selected by both players, it activates the fire particle system.	30
3.9	The puzzle level with the barriers, as well as the buttons the players use to open them.	33
4.1	Model of the excavator mining machine provided by the mining department at University of Nevada, Reno.	40

4.2	Model for the mining drill rig, retrieved from [11].	41
4.3	This image displays the use case diagram of the proposed software.	48
4.4	This image displays a block diagram of how the system is supposed to function. The operator uses a VR headset and HTC-Vive controller to control the machine. The VR headset connects directly to the UI, whereas the Joys-stick connects to the locomotion prefab, which can alternatively control the mechanical aspects of the machine. The camera objects read in video input and send it to the machine UI.	50
4.5	This picture shows the design class diagram of how the software will be constructed.	51
4.6	Each of the four frames are merged together and stored in a single variable, which is then used as the output frame.	54
4.7	This image shows the result of merging four webcams together and then loading that resulting frame to a web server.	54
4.8	The result of recording a video clip of the panoramic web page and using it as the skybox background in Unity.	55
4.9	The C # script used to load in a live webcam feed directly into Unity and render it to a polygon.	56
4.10	The result of loading the webcam feeds directly into unity and attaching them to polygons in order to create a panorama.	57
4.11	This is an image captured of the scene view in Unity of the prototype. Here you can see the mining machine model with the camera objects attached to it, along with the render textures set up in an order to create a panoramic frame.	58
4.12	Here is a snapshot from the game view window in Unity of the prototype. This shows the viewer within the skybox, which is meant to display what the user sees out of the camera objects placed on the machine.	58

Chapter 1

Introduction

Virtual reality (VR) is a technology that has been around for decades, and within the last decade alone, has gained a massive amount of popularity. This is in a large part due to the technology's separation from being exclusively for research labs, and integrated into the homes of millions of people. This is not to say that the popularity in the research community has subsided as a result of VR becoming more widespread in the mainstream, on the contrary, it has increased the interest in the subject tenfold. There are now a host of many different research institutions that are taking an interest in VR technology as a result of its widespread popularity, ranging from fields such as game design to robotics. This thesis outlines three research areas related to VR in order to exemplify the diverse subjects that are tied to the field. The first two areas represent the areas of VR that are interesting and which helped build a foundation for understanding VR, whereas the third area steps out into a real-world application, and finally Chapter 5 presents conclusions and future work in VR related to these areas.

The first topic this thesis discusses is a user study that was conducted in order to analyze two types of locomotion in a virtual environment (VE). This topic is covered in Chapter 2, and this chapter also appears as a conference paper in ITNG 2020 [2]. The two locomotion techniques this study investigates are a natural and inorganic method, where the natural consists of a walking in place (WIP) mechanism and the inorganic is comprised of using an HTC-Vive wand to induce movement. The participants in the study are placed in a VE designed as a walking track with numerous checkpoints

that the user must cross until they reach the finish line. Factors such as efficiency and time are measured for each form of locomotion, as well as factors that influence the individual's preferred locomotion technique.

The second topic that is covered in this thesis is the creation process of a multi-user cooperative VR game in Chapter 3. This chapter outlines the steps it takes to create a multi-user puzzle game using the Unity game engine [13] and the Blender modeling tool [4] such as level design, networking, and developing the models. This chapter is also featured in ITNG 2020 [6] as a conference paper. The two users play as witches who are equipped with a scroll that allows them to access the powers they use to solve puzzles. The powers are placed around the map and appear in the form of circular orbs, the players must collect these power orbs in order to access them on their scrolls. The levels are designed in such a way that require the players to think methodically by working together in order to complete the challenges. The data structure that is used to store the powers in the Unity script is a queue, which is only accessed when the players press the power buttons on their scrolls.

The third and final topic of this thesis in Chapter 4 describes our contribution to the Immersive Teleoperation of Mining Machines (ITMM) project, which entails combining live-stream video feeds into a single panoramic frame so that it can be viewed in an HTC-Vive headset. This chapter is also slated to be featured in a future conference. Two methods are used to merge the live-feeds of multiple webcams together into a panorama. The two methods are then compared to each other in order to discern which one would be most effective in a real world setting in terms of video quality. The first method uses OpenCV and the flask web framework for Python to merge the videos together into a panorama, and then send the merged frame to a web server. From the web server it should then be loaded into Unity to then be applied as a skybox background. The second method uses a C# script to load footage from the webcam live-streams directly into unity, and then renders the footage to a polygon. It is then up to the programmer to place the polygons next to each other to manually create a panorama.

Chapter 2

Locomotion in VR: A Comparison Between a Natural and an Inorganic Locomotion Technique

This chapter first appeared as a Conference publication in ITNG 2020 [2].

Kurt Anderson, Lucas Calabrese, Andrew Flangas, Sergiu Dascalu, Frederick C. Harris, Jr. (2020) A Comparison Between a Natural and an Inorganic Locomotion Technique. In: Latifi S. (eds) 17th International Conference on Information Technology–New Generations (ITNG 2020). Advances in Intelligent Systems and Computing, vol 1134. Chapter 42, pp 317-323. April 6-8, Las Vegas, NV. Springer, DOI https://doi.org/10.1007/978-3-030-43020-7_42

Abstract

Virtual reality is becoming a more popular attraction every year not only to researchers, but the general public as well. One of the major challenges standing in the way of virtual reality becoming even more widely accepted is the adaptation of new locomotion techniques. This paper attempts to discern between two different locomotion techniques and decide which method is more efficient based on certain parameters. The two techniques being analyzed were tested in a case study, one involving inorganic movement (touch pad control) and the other natural movement. The users tested both forms of locomotion separately by navigating through a pre-

determined course that is comprised of multiple checkpoints. Data such as efficiency and time were recorded via applications, as well as a post test survey that each of the participants were given. After all the data was collected, the results were analyzed and the most efficient and preferred form of movement was established.

Keywords: Virtual Reality, VR Motion Sickness, Locomotion, Inorganic Movement, Natural Movement

2.1 Introduction

The purpose of this study is to compare a natural and an inorganic method of locomotion in Virtual Reality. The study tested to see which method is superior when utilized by a sample of college students. At the least, we explore which method allows the users to traverse the virtual environment more efficiently. Natural methods are those that mimic something that the human body is already used to [15]. The natural method in this study is a walking in place (WIP) method, in which the user moves forward in the virtual world at a fixed pace as they walk in place. The inorganic method uses the touchpad on the HTC Vive controller to move the player forward. The touchpad movement, will control the users avatar in a way that the body has to learn. Another example of a learned method is a person using a mouse with a computer. When first introduced to a computer, moving the mouse on an x,z plane reflects to the mouse moving on an x,y plane on the computer [15]. One of the main focuses of this study is to determine if natural and inorganic methods correlate to efficiency and immersion in virtual reality. This method also moved the user at a fixed speed in the Virtual Environment (VE). In both methods the user moves in the direction the head-mounted display (HMD) is facing.

This study is important due to its emphasis on the topics that are related to the limitations of interacting with VR. While natural walking in room scale is available for Virtual Reality, it does not provide users with the ability to walk outside of the play area. Because of this and the many games that require different types of locomotion, there are multiple types of locomotion being researched in order to find a method

that includes certain requirements. These requirements include not being restricted by the size of the play area, allowing users to travel far distances with reduced fatigue, and meets the requirements of the game [1]. It is also important to ensure that the method does not make the user feel ill. This gives us reasons to compare the natural and inorganic methods to find out which is superior.

Developers will benefit from this study depending on the type of game being developed. This study can give developers an idea of which method is more efficient, and which method meets their needs. If immersion is important, WIP may be chosen for a method of locomotion. If a potentially less taxing method of locomotion is required, they may choose the inorganic method for their locomotion requirement. Developers may even be able to utilize both as users may have a preference between which method they would like to use. This study may show which method is superior when it comes to cybersickness, efficiency, fatigue, and immersion. Cybersickness and fatigue are examined using questionnaires, while efficiency is determined through the time it took for players to navigate through the VE.

The rest of this paper is structured as follows: Section 2.2 outlines the implementations of the natural and inorganic methods. Section 2.3 describes the participants, the hardware used, and the design of the study. Section 2.4 covers the results of the experiment, and the responses for the feelings of fatigue and cybersickness. Section 2.5 includes discussion on the implications of the data and responses. The conclusion and a discussion of future work is presented in Section 2.6. The last Section 2.7 contains the acknowledgements.

2.2 Implementation

Both implementations use the forward direction of the HMD to determine which direction the user will be translated as movement occurs. It was possible to make the direction of movement for the inorganic method dependent on the controller, but we felt it added unnecessary complexity. Both methods multiply the translation vectors by `Time.DeltaTime` [31] to ensure that the speed of the computer does not affect the

speed of the user. All the checks for input were in the Update function [31].

Code was developed for detecting motion for WIP movement. In this code the height of the feet is compared to check if one foot is more elevated than the other. For the inorganic method, the touch pad was monitored and return values had to be towards the edge of the touch pad in order to move. This helped make the movement smoother and eliminated jerkiness while the thumb moved in the middle.

2.3 User Study Methodology

In order to increase the overall understanding of the data collected from the study, all participants were asked to fill out a pre-test survey. The survey asked questions about topics such as their current energy level, as well as their experience with VR and video games. A post-test survey was also administered to gather information on how the users felt about the different locomotion methods. The post-test survey questions about feelings of fatigue, sickness, and overall enjoyment. The users experienced both methods. The order of the methods tested were randomized in order to reduce the odds of the data being affected by the ordering.

2.3.1 Participants

The participants used in this study consist of a wide and diverse range of people, both of different backgrounds and genders. There were a total of 20 participants in which 12 were male and 8 were female. They were all at one point enrolled at a university, with fields ranging from marketing to computer science and engineering. We made certain to choose more participants with no background in computer science in order to properly gauge how people outside of the field react to virtual reality. By choosing participants outside of the field of computer science and engineering, it also increased the chances that the subjects never used virtual reality before.

Most of the participants answered in the pre-test survey that they had little to no experience with virtual reality. A higher percentage of males answered that they had little virtual reality experience compared to the females. It is also worth

mentioning that two of the female participants were two out of the three computer science and engineering majors used for the study. Most of the participants, both male and female, also claim to have had experience playing video games with the exception of a few that answered little no experience. There was also about an equal amount of participants that answered in the survey that their energy level was either high or low/moderate.

Most of the participants answered in the pre-test survey that they are not prone to motion sickness; the majority that answered yes were female. In the post-test survey, both male and female participants answered that they felt more motion sickness, as well as fatigue using the natural method. As a result of this, most of the participants in the post-test survey answers suggest that they prefer the inorganic over the natural method of locomotion. In the post-survey, there did not seem to be too much of a disparity between the male and females involving their level of motion sickness and fatigue. A majority of the participants answered that participating in this study increased their interest in virtual reality.

2.3.2 Apparatus

In order to avoid preventing players from being faster or slower in either method, both the organic and inorganic methods used set speeds. However, the natural methods could have been allowed to have varying speeds if deemed necessary.

We used the Unity Game Engine in order to create the application in which we had participants use (Fig. 2.1) [31]. Within Unity, the SteamVR asset package as well as the Hand Painted Forest Environment Asset shown in Fig. 2.2 [8, 39] was used. These allowed us to create an aesthetically pleasing environment for the participant to move through, as well as provide us with a library to interface the HTC Vive headset with.



Figure 2.1: A perspective view of what participants would see when the testing application starts

2.3.3 Procedure

The participants were then introduced to the virtual reality system and the space that they were going to be spending the duration of the study in. Then the participants were given the pre-test survey to fill out. Some of the questions asked were about their experience with VR, their reason for coming, and what their major is. After completing the survey, the first locomotion method to be tested was explained to them. Once they confirmed they understood the method, they were given the headset and controllers. The participants were told that if they felt sick they could stop the testing at any time. Then they familiarized themselves with the method by exploring the VE before the test was started. The users then completed the course and data was taken.

After completing the first method, the users were given a five minute break. The break was given to reduce the odds of fatigue, along with the possible feeling of motion sickness so that results from the next method were not affected. After the break, they were briefed on the next method. When they verbally confirmed that

they understood the method, they were given the headset and feet sensors. They were given time to get used to the method of locomotion, and completed the course again. The data was again recorded. Finally, after obtaining all of the data, the users were then asked to fill out a post-test survey. The survey asked questions about which method was more immersive and enjoyable. The participants were also asked to gauge how sick or tired they felt on a scale of one to ten.

Throughout each run of the course, two separate times were measured. The most important of the data collected was the overall time from the starting line to the finish. The time it took for the user to move between each checkpoint was also recorded. The data was saved as split times for easier data analysis. The final pieces of data that were collected are from the post survey. These were just a hard value given as an opinion of how motion sick and fatigued the participant felt.

2.3.4 Tasks

As stated previously, after the participants were briefed on the control scheme of each locomotion method, they were allowed one minute of free time in the virtual world to explore using said method of locomotion. Once their exploratory time had expired, the participant was moved to the starting zone of the application. Once they moved through the first checkpoint, the timer was started and they began moving through a predetermined course as fast as possible using the respective locomotion method. There was a single path on the ground for the participant to follow as seen in Fig. 2.2. In order to complete the course, the participant needed to navigate through a series of checkpoints. The current checkpoint the participant had to reach was seen as a massive translucent green screen. Once the user reached the end of the course, they were briefed on the other method of locomotion and followed the same steps.

Throughout each run of the course, two separate times were recorded. As mentioned earlier, the most important was the overall time from the starting line to the finish. The time it took each user to move between each checkpoint was also recorded. These values were saved as split times for easier data analysis. The final pieces of



Figure 2.2: An overview of the course for each participant to follow.

data that was collected was from the post survey. These were just a hard value given as an opinion of how motion sick and fatigued the user felt.

2.3.5 Design

In terms of variables for this study, there were not any between-subject variables. Our independent variables were all within-subject. The independent variables were the style of locomotion and how sick and fatigued the participant felt at the end of each course. Each participant performed the two styles of locomotion. The order in which the participants performed them was random. This way, the data was able to be gathered in a more efficient fashion. If each participant did one style first and the second after, then the data could be skewed towards the second movement style being more efficient. This would be because the participant would already know the course. The overall entry for this study was 20 participants, two forms of locomotion, and one course to move through. The course contained sixteen checkpoints, including

the time between the last checkpoint to the finish.

2.4 Results

2.4.1 Course Data

Fig. 2.3 displays a box and whisker representation of the overall lap times for each participant. We used a One-Way ANOVA calculator to find the p-value [29]. The p-value that resulted from the data was .024. Thus, the data between the two populations is statistically significant. The averages between each participant for each reached checkpoint were very close. However, the average times it took for each participant using the inorganic method to reach each checkpoint were faster than the average times it took for each participant using the natural method to reach each checkpoint.

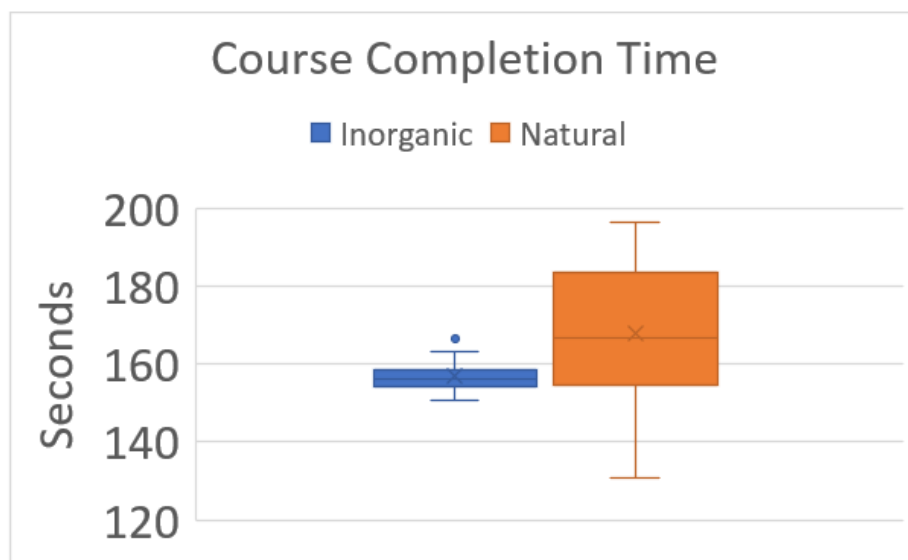


Figure 2.3: Box and Whisker Plot for the course completion times of each participant in seconds. **Inorganic:** Average = 156.8, Median = 156.2, Outlier = 166.6, Maximum = 163.4 Minimum = 150.7 **Natural:** Average = 167.7, Median = 166.8, Maximum = 196.3, Minimum = 130.6

2.4.2 Cybersickness Data

Fig. 2.4 shows the Box and Whisker Plot for feelings of motion sickness. We used a Likert scale from 1 to 10 to collect data on feelings of cybersickness. In the scale, 1 means that the user felt no symptoms of motion sickness, while 10 means they felt extremely sick. After running the data through a one-way ANOVA calculator[29], the p-value between the two populations resulted in .069. If $\alpha = .05$ the data between the two populations is not statistically significant. A fair number of participants, one half, felt no sickness whatsoever using both methods. Some felt the inorganic method caused more sickness while some felt the WIP method caused more sickness.

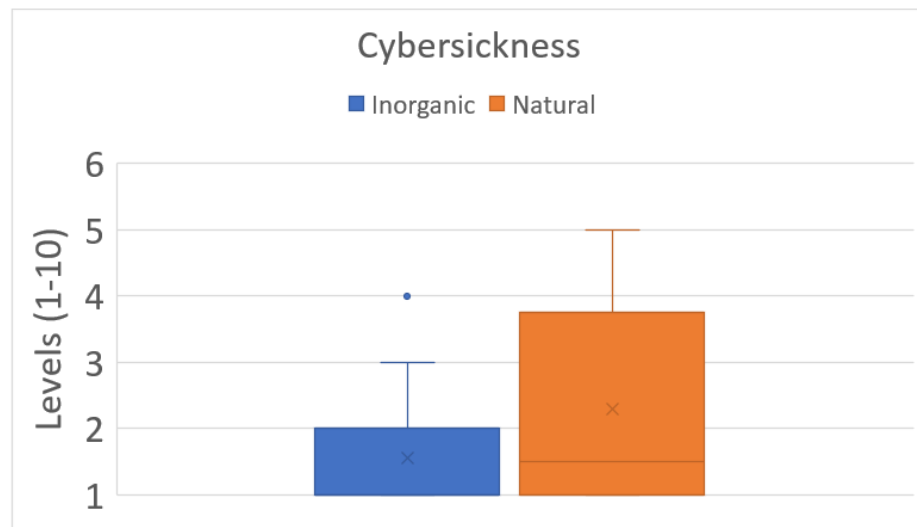


Figure 2.4: Box and Whisker Plot for Cybersickness Responses. **Inorganic:** Average = 1.55, Median = 1, Outlier = 4, Maximum = 3 **Natural:** Average = 2.3, Median = 1.5, Maximum = 5

2.4.3 Fatigue Data

Fig. 2.5 shows the Box and Whisker plot for feelings of Fatigue. When we gathered data for fatigue we used a Likert scale from 1 to 10. In this scale, 1 means the participant felt not tired, while 10 means they felt extremely tired. We found the p-values for the data received for fatigue using the same calculator [29]. The p-value is $< .00001$. We had trouble interpreting the responses from ID 16. This is because 16

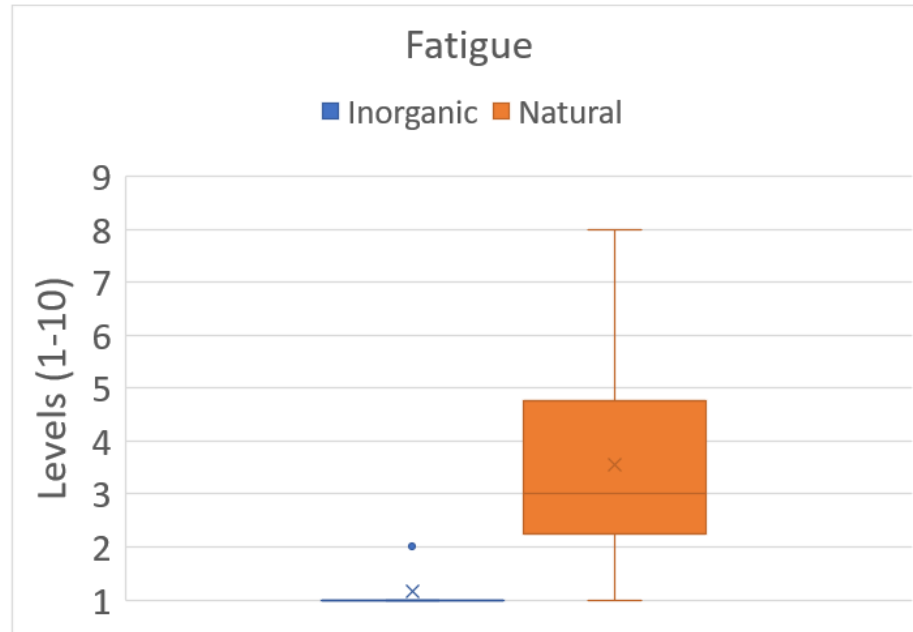


Figure 2.5: Box and Whisker Plot for fatigue responses. **Inorganic:** Average = 1.15, Median = 1, Outlier = 2, Maximum = 1 **Natural:** Average = 3.55, Median = 3, Maximum = 8

answered that the inorganic method was more tiring because of walking. We assumed this was a mistake and swapped that participant's values for fatigue.

2.5 Discussion

2.5.1 Comfort

According to the post test surveys, most of the participants reported the inorganic method as being more comfortable with the exception of three. One of the three users that preferred the natural method mentioned that they got tangled up in the wires from the headset during the inorganic method. This probably affected their decision to choose the natural method. The other two reported that they preferred the natural method because they found it more entertaining.

There were some common trends that appeared with the participants that chose the inorganic method as being more comfortable. These trends included the users being able to stand still while moving as compared to walking in place, the ability

to turn easier, and overall less physical exertion. However, there was one participant that claimed they preferred the inorganic method when walking a straight line, but they found it more comfortable to make turns using the natural method. As mentioned earlier in the participants section, the users also found the inorganic method to be more enjoyable. Perhaps there is a correlation between comfort and level of enjoyment, despite the natural method being more immersive.

2.5.2 Intuitiveness

One participant claimed to have felt that the inorganic method was more intuitive due to having played video games in the past. At least two participants stated that the inorganic method was more intuitive due to having experience with VR. Some participants felt that the controller method was more intuitive because of the feeling that they had more control, while at least one stated that it was more natural to them. One participant said "... I felt like the way I was walking felt more strange than just guiding with my hands." However, several felt that they had more control walking in place whereas others felt that walking in place was more natural. One participant said "the organic method was more intuitive because it was [the] easiest to control directionally, as if I were actually walking." Another said, "I thought the foot sensors felt more intuitive because the movement felt more natural."

2.5.3 Immersion

A few participants reported that the inorganic method was more immersive while the majority felt that the natural method was more immersive. Many felt that the walking was more realistic, since they were actually moving their legs rather than just using a controller. One participant said the inorganic method felt more immersive because "it felt as if [the participant] was actually moving through the simulation rather than just gliding through it." However, there was one participant that felt that the inorganic method was more immersive because the participant "did not have to think about walking as much." Based on the responses, the natural method appears

to be superior when it comes to higher levels of immersion.

2.5.4 Cybersickness

Cybersickness is a condition that likely occurs in certain individuals that may be caused by conflicting senses like viewing movement through eyes, while other cues such as vestibular cues tell the brain that there is no movement.[18] One of the main aspects that keeps VR from becoming even more widely accepted is cybersickness or VR motion sickness. Another important factor that applies to developing a successful locomotion method is the level of VR motion sickness it induces. This part of the study focuses on which locomotion method induced the most VR motion sickness. The intention for this subsection is to discuss how participants felt about both methods.

Participants were asked about the level of motion sickness they felt using both methods in the post-test survey. Several stated that they felt sick from the inorganic method while making turns, “there was a little motion sickness because of turns.” A participant that felt sick from the WIP method said: “Because I was actually moving, there were times where I would turn and almost fall over, it felt as if my body was moving faster than my legs.” It is possible that this could be due to the speed of the simulations as the users moved somewhat fast in the VE. Another said: “I felt like I was losing my balance a few times and my stomach kept flipping.” But the same person said: “[I] Only got a little disorientated going around sharp corners” when talking about the inorganic method.

We were surprised that users felt sicker from the natural method. There were 9 people who reported to not feel sick at all, which may be why the medians are so low for both methods. Overall, at least according to the averages, WIP seemed to cause more feelings of sickness. It also seemed to cause more intense feelings of sickness to users who are affected by cybersickness.

2.5.5 Fatigue

As expected, the majority of users found WIP to be more physically exhausting than using the touchpad. Some felt that the difference between the physical exertion from the two methods was large, while some did not seem to really notice it at all. It is worth mentioning that ID #20 said that he/she was feeling tired that day. Some responses for how users felt how tiring the natural method was includes: “Slightly Winded”, “more movement involved, as if I was exercising”, and “I didn’t expect to exert as much physical energy as I did.” Most felt little to no physical exertion from the inorganic method.

The inorganic method does not require much effort, which is why there is almost no variability. The natural method on the other hand did have noticeable variability. This could be due to different effects of exercises on participants and because the speeds vary due to the many different ways users can give input for walking.

As mentioned in the participants section, there was an equal amount of users who reported high and low/moderate levels of energy before participating in the study. Even most of the participants that reported having high levels of energy still said that the natural method was more physically strenuous. There were a few exceptions who reported not feeling tired at all after testing the natural method, one participant stated “It didn’t require noticeable physical exertion.” Therefore, it is safe to conclude that the inorganic method is better in terms of lower levels of fatigue. In a later section, we discuss how this observation could have affected the user’s level of enjoyment for each method.

2.5.6 Efficiency

According to the average time it took for users to complete the course, the users were faster using the inorganic method. One user said, “I thought the controller-based movement was the most efficient because I was able to move at the same speed throughout the test. Also when making turns when using foot sensors, it felt disorientating.” Another said “I think the inorganic method was most efficient in terms of

speed. It also required the least amount of movement. However, moving felt more difficult to control (directionally).” Several others mentioned that they felt the inorganic method was more efficient due to it being easier to use than natural. Another felt that the smoothness of the inorganic method was preferable, “The inorganic method because it was just smoother overall and it provided continuous in-game movement.”

When it comes to the averages times of how fast each checkpoint was reached, the natural method took a little longer for each checkpoint. The parameters were adjusted so that one of the authors could reach the finish line at similar times for both methods. The parameters may have had a decent effect on the results. According to the data for lap completion, the inorganic method seemed to be more efficient. The natural method was also more complicated to use which may have had an effect on speed. This is because users would have to figure out how to walk in place in a way that allowed them to move smoothly without stopping. Although it may seem that WIP should feel more natural to users, some have commented on it not feeling natural.

While some felt that the inorganic movement was faster, others felt that the natural movement was faster. One participant said, “the controller method was a lot easier, but I feel the foot controllers were faster.” A participant said, “the most efficient method to me was the natural method. I felt more grounded. when I had to turn my body using the inorganic method, I found my balance to be poorer.”

While the data shows that the inorganic method was more efficient, this could have been affected by the parameters used for the methods. This might be partly why users felt that the touchpad method was more efficient. Around five users felt that the inorganic method was more efficient due to it being easier to use, or because it did not require as much energy to move in the VE. The rest of the participants gave other reasons for why they thought the inorganic method was more efficient.

The natural method had more variability than the inorganic method. This may be due to the inorganic method being much simpler to use. The natural method required walking in a certain way to move as smoothly as possible in the VE. The inorganic method did not require much skill to move at a constant speed. This may be

why there is not much variability with the inorganic method, but the natural method has participants who completed the laps quickly, and some who completed the laps slowly.

2.5.7 Enjoyment

A majority of participants reported that the inorganic method was more enjoyable than the natural method. Several have stated that they enjoyed the inorganic method more due to feeling less cybersickness. Others felt that the inorganic method was more enjoyable because it required less physical exertion. When asked the question about which method was preferred, the participant said, “the inorganic method because I’m lazy and didn’t enjoy picking up my feet to get through the course.” It seems that fatigue may have been a major deciding factor for this part of the study. It also seems that the excitement of immersion in this case, did not outweigh the user’s dislike of feeling fatigued.

However, the few participants that enjoyed the natural method gave reasons that included the enjoyment of walking itself or for the immersion. For example, one participant stated “Natural was more enjoyable because I felt more engaged with the environment.” Another participant said “I found the foot sensor usage was more enjoyable. Although the turning felt a little disorientating, it still felt fun to actually walk in a simulation. It felt like I was in a footrace.”

2.6 Conclusions and Future Work

2.6.1 Conclusions

In this study, we have found the inorganic method to be the preferred method of locomotion within our sample. This is due to the responses received for feelings of fatigue, cybersickness, enjoyment, and comfort. We also found that the users were more efficient with the inorganic method, using the speed of the average times for completing the course as the metric. However, the WIP method was superior

pertaining to feelings of immersion.

2.6.2 Future Work

This case study may have compared the inorganic and the natural methods when walking through a path, but it does not necessarily show how the methods are affected by situations in a real game scenario. However, a similar scenario as the simulation could be related to racing games, but not all games will require the player to travel a long distance without taking a break. Some games like puzzle games for instance, may only include some walking but with less continuous movement. Puzzle games may expect users to keep track of a lot of information or to be in the process of puzzle solving. The different effects of cognitive load provided by each method may have an effect on gameplay. This means that whichever method is superior depends on the type of game being played. Therefore, there could be more research done looking into these methods based on different tasks rather than walking in a lap. This research should include a better analysis of the effects pertaining to cognitive load in their respective scenarios.

The locomotion methods tested in this study are just two of the many types of locomotion methods. Another interesting method is redirected walking. Although redirected walking requires at least a 6x6 meter play area, it is an interesting method for ensuring that the user can continuously walk forward in the VE without leaving the play area [17]. An inorganic method that was not looked into is teleportation. Teleportation is where users can instantaneously move to a new position in space. This teleportation can be implemented in a number of ways. Some can include activation of the teleportation by use of the controller, blinking, stomping, or looking at a position for an extended period of time [28]. There are many more possibilities and methods. More studies can be done looking into these methods to try and find which will best serve the user's needs.

2.7 Acknowledgements

This material is partially based on work supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

IRB Approval under IRBNetID: 1487456-1.

Chapter 3

Multi-player Game: Multi-User VR Cooperative Puzzle Game

This chapter first appeared as a Conference publication in ITNG 2020 [6].

Lucas Calabrese, Andrew Flangas, Frederick C. Harris, Jr. (2020) Multi-User VR Cooperative Puzzle Game. In: Latifi S. (eds) 17th International Conference on Information Technology–New Generations (ITNG 2020). Advances in Intelligent Systems and Computing, vol 1134. Chapter 39, pp 293-299. April 6-8, Las Vegas, NV. Springer, Cham, DOI https://doi.org/10.1007/978-3-030-43020-7_39

Abstract

Multi-user virtual reality (VR) games are at the cutting edge of interpersonal interactions, and are therefore uniquely geared towards real-time interactive games between human players. This paper describes the process of designing a cooperative game where the obstacles are designed to encourage collaboration between players in a dynamic VR environment. This is done using the Unity game engine and the Blender graphics modeling tool. We demonstrate the progress of our scheme in a multi-player cooperative game, as well as the importance of the VR interface for encouraging cooperation. The VR experience provides a realistic human-human interaction improving on generic game-play, as our system utilizes the real-time interface to create an entertaining VR experience.

Keywords: Multi-player, virtual reality, Real-time, interactive, Unity

3.1 Introduction

The advancement of VR technology has opened the door to many different possibilities considering the numerous applications for it, one of which being gaming. To explore how VR technology can be used in multiplayer games involving a virtual environment (VE), this paper will discuss the process of designing a two-player cooperative VR game. This game was customized for the HTC Vive headset and the Steam VR software. The game was designed for two players to work together to overcome obstacles. Teamwork is not only encouraged, it is required if the players wish to successfully advance through the levels.

Games such as this will encourage multi-user VR scenarios [37] and create a more social atmosphere for players to enjoy. In this game, the players utilize different powers that come in the form of crystal balls that can be picked up, and that are placed strategically throughout the game world in a way that the players will have to make use of their problem-solving abilities to reach them. Once the powers have been obtained, the players will have to use them in a specific way to solve the current obstacle in front of them. Multiple improvements can be made to make the game better as a whole that is described later in Section 3.2.2, but due to time constraints, these improvements are not present in the prototype version of the game.

The rest of this paper is structured as follows: The Creation Process is described in Section 3.2. Gameplay is presented in Section 3.3, and Conclusions and Future Work are covered in Section 3.4.

3.2 The Creation Process

3.2.1 Blender Modeling

The initial stages of the creative process involved developing the models for the game using the open-sourced 3D computer graphics software toolset Blender [4]. The witches were constructed by molding two mirrored cubes together to create the torso and then the rest of the body. Other objects were attached to the body to create the

arms and shoes. Blender’s bezier curves were used to create the hair of the witches and were set as children of one of the bones after being imported to Unity [13].

The next step was to create the animations for the witches. One of the better animation papers was written by Narang, Best, and Manocha [23]. A basic algorithm has been implemented into Blender using the Rigify add-on to use a human rig. Our model, the Rigify, and animation controls can be seen in Fig. 3.1.

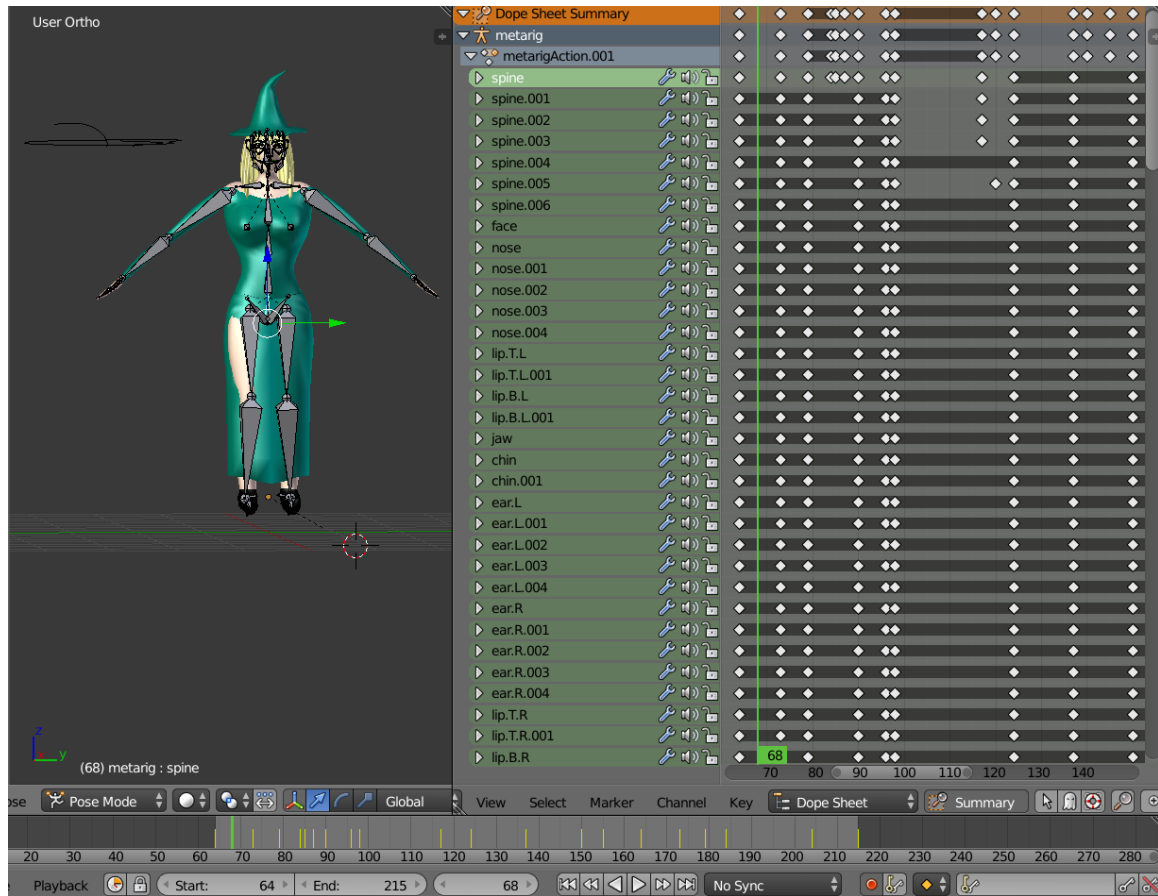


Figure 3.1: A Screenshot in Blender which shows how to setup animations using the human rig for the witch model.

Automatic weights were used for the animations, but some adjustments were made with weight painting. The animations for the witches included walking forward and backward, sidestepping, and jumping. The arms were intentionally not animated for walking so that they could be controlled with Inverse Kinematics. Once the two witches were created, they could then be used for the initial stages of the development

of the game in Unity. A scroll to act as the selection menu for the powers was also created by molding a single cube. The next models to be designed were the crystal balls that the witches collect and use in the game. The crystal balls were comprised of transparent sphere objects with an animated object in the middle that represents the power that it grants. To go along with the crystal balls were the crystal ball stands to keep them in place and to spawn them. The crystal balls and their stands can be seen in Figure 3.2



Figure 3.2: Crystal Balls representing powers on their Stands

Later in the developmental stages of the game, Blender was used once again to design an octopus-like creature with four tentacles and a water projectile for it to shoot at the players. The octopus started as a single-cylinder that was molded into the shape of the head, and then four mirrored cubes were used for the tentacles. Blender's Inverse Kinematics was used to make tentacle animations. It was given idle, walking, and attacking animations. The head was given a bone so it could look up and down, while the entire model rotates to face the player. The water projectile was also given a rig to create the animation of it swelling and bubbling like a ball of water. Later in the development process, the levels were designed in Blender and then imported into Unity. The final model can be seen in Fig. 3.3.

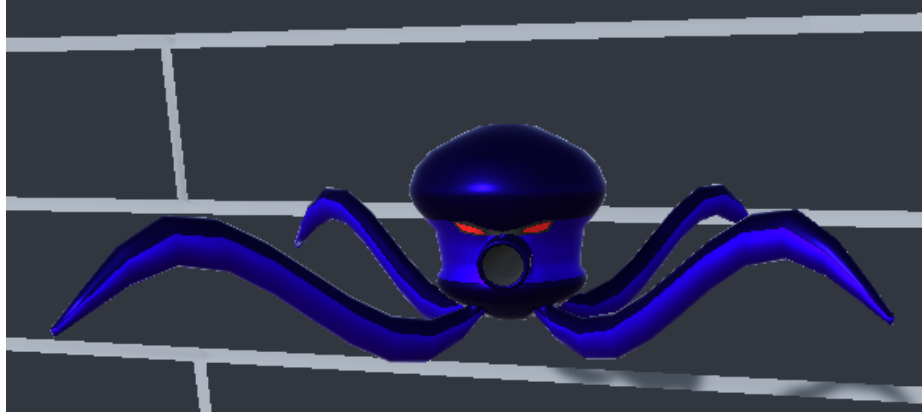


Figure 3.3: The Octopus model

3.2.2 Development in Unity

To test the powers that the witches use, as well as other gameplay features, a sandbox was created with a single plane as the floor of the scene with four walls surrounding it. The first object created in the scene aside from the planes and walls was the player prefab. The player prefab consists of a VR camera along with a right and left-hand object.

Then it was time to attach the scroll to the transform of the right-hand controller. The transform of the controller was used so that it can be rotated more freely than if it was attached to the model's hands. The purpose of the scroll is to act as a selection menu for whichever power the player wishes to use, as well as keep track of the number of powers the player has picked up. For buttons that activate powers, the scroll used models of the crystal balls that were scaled to look like buttons. A box collider was used for each of the witch's hands to register when the hands were touching a button. Text Mesh Pro was used to display the amount of each of the crystal balls collected by the user. Two more buttons were added to the scroll, a stop button to cancel any power currently being used, and a swap-hands button for the scroll so right or left-handed people can choose the setting that is most comfortable for them.

A script was used to fix the model's position slightly behind the camera. The

character controller that is used for detecting collisions adjusts its central location to keep all players at the same height regardless of their height, or whether or not they are sitting down. Cloth was used for the witches' dresses, in which capsule colliders attached to the model's bones were selected to allow them to collide with the dress. Because the cloth would get stuck on the colliders, a script was added to reset the cloth under certain circumstances such as when the model jumps.

3.2.3 Developing the Powers

After the Inverse Kinematics and scroll were set up, it was then time to focus on the coding of the powers. It was decided that there would be five powers: swap, shrink, freeze, bomb, and a fire power for this prototype. In order to obtain a power, the user must find and have their player touch a crystal ball representing that power.

All five powers and their associated effects are illustrated in Fig. 3.4-3.8.

Swap Power: The swap power (Fig. 3.4) is used to instantaneously switch the position of the players with other GameObjects in the scene.

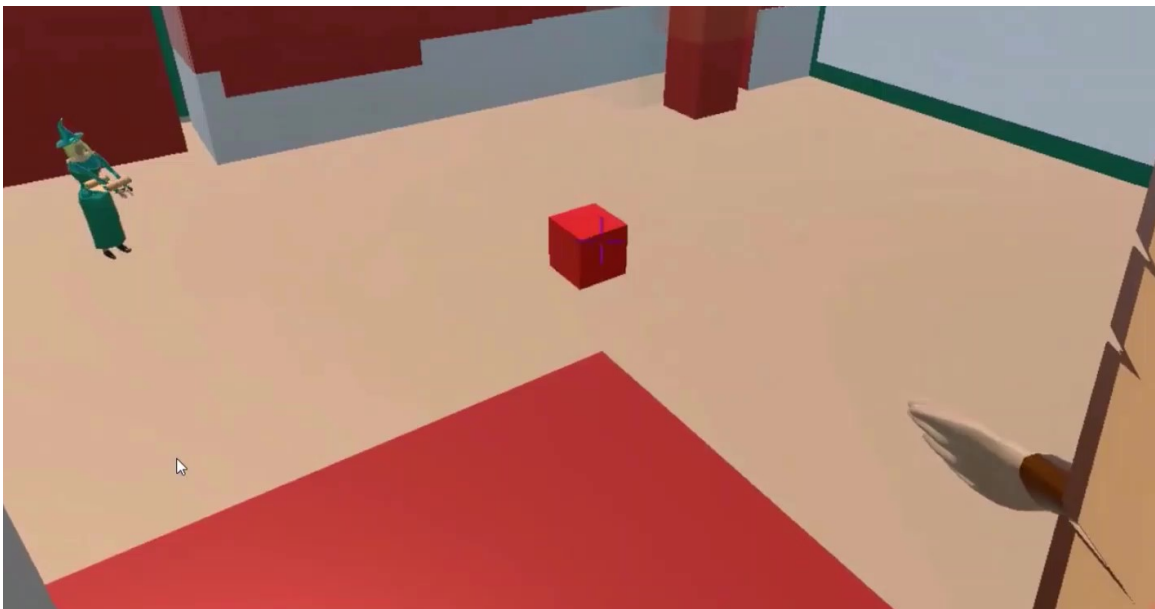


Figure 3.4: A first person view showing the object that the swap power applies to has its material changed to red. After the swap power is applied, the player will switch places with the cube.

Shrink Power: The shrink power (Fig. 3.5) is designed for the players to fit through small tunnels or other similar obstacles by making the player significantly smaller.

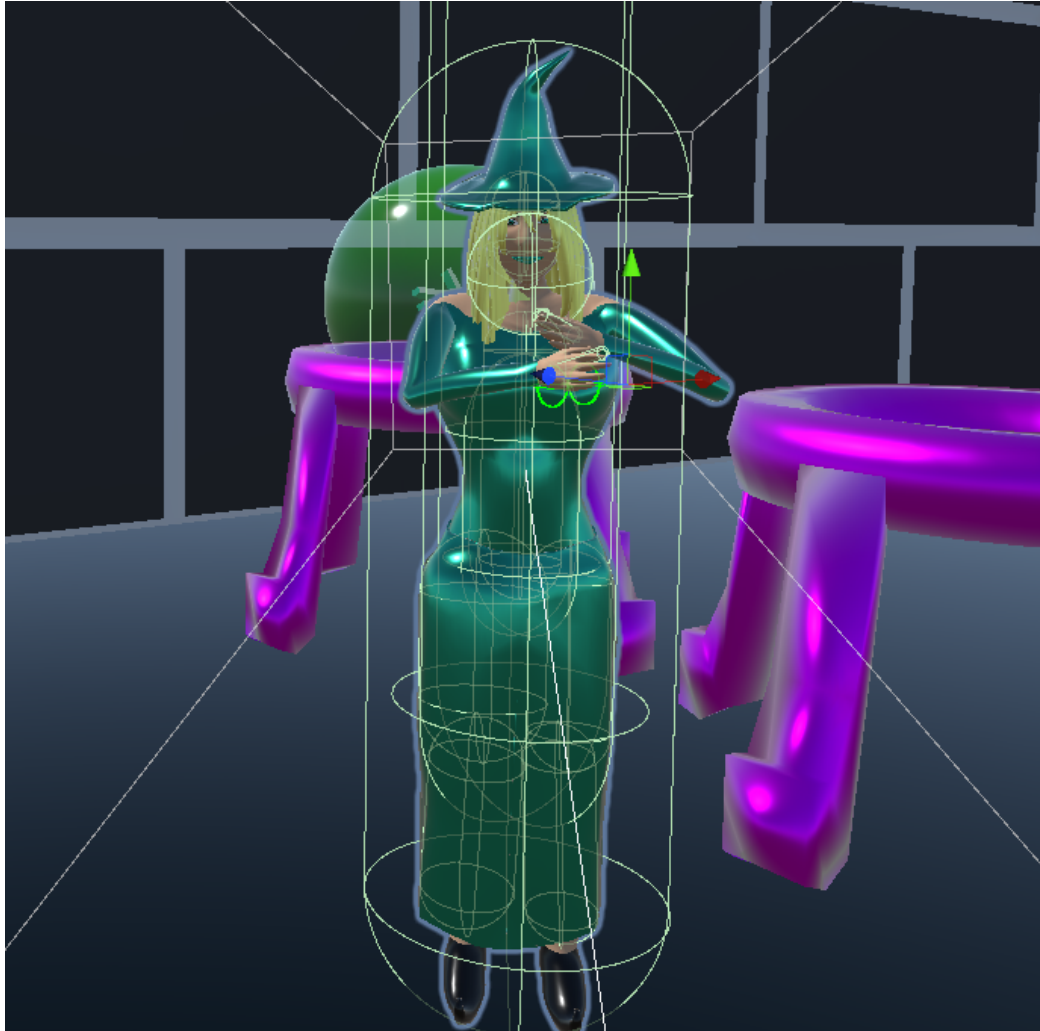


Figure 3.5: The shrink power scales the player to a considerably smaller size. Notice the crystal ball stands next to the player.

Freeze Power: The freeze power (Fig. 3.6) is used to turn the water projectile the octopus shoots at the player into a cube of ice, and then to use the cubes of ice as a jumping platform. This was meant to encourage teamwork as the octopus would follow one player around and shoot a bubble of water at that player and when it is turned to ice the other player who can use it as a platform. An additional purpose that

was added to the ice power was causing a balloon object to descend when activated due to the change in the balloon's volume due to its cold temperature.

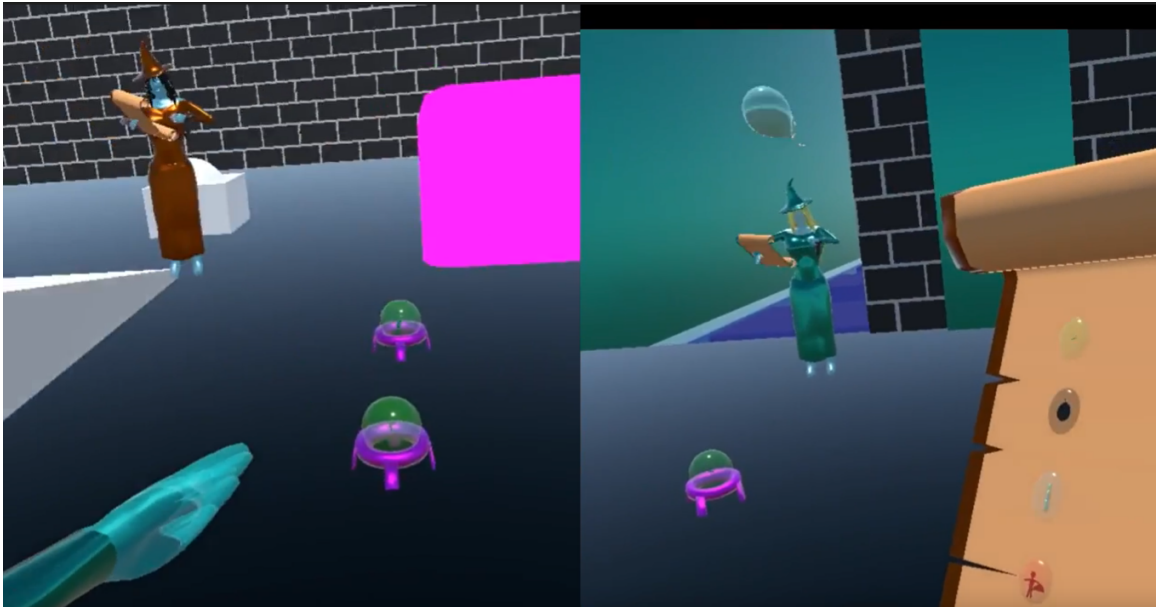


Figure 3.6: A split screen (two images from different player's screens). When the ice power is selected and activated it changes the material of your witch's skin into a light blue color. The ice power is bringing down the balloon seen in the right window.

Bomb Power: The bomb power (seen in Fig. 3.7) appears in the hand of the player when selected. The player can then grab and throw the power at something else in the game. This bomb power then explodes on contact.

Fire Power: Lastly, the fire power (Fig. 3.8) is designed to melt the already frozen cubes that are created using the ice power and cause balloons to ascend. Some functionality could still be added to the ice and fire power to give them more use.

To make these powers accessible to the players via the scroll, a powers script was created and attached to the witch `GameObject`, which was a child of the player object. In this script, each of the powers are stored in a queue, and only accessed when the player presses one of the buttons. The queue stores crystal balls. These objects are returned to their stands either 7 seconds after use, or if they are away from their stands for 7 seconds. `GameObjects` were stored to make the transforms of the crystal balls and the `setActive` function easily accessible. It was essential to

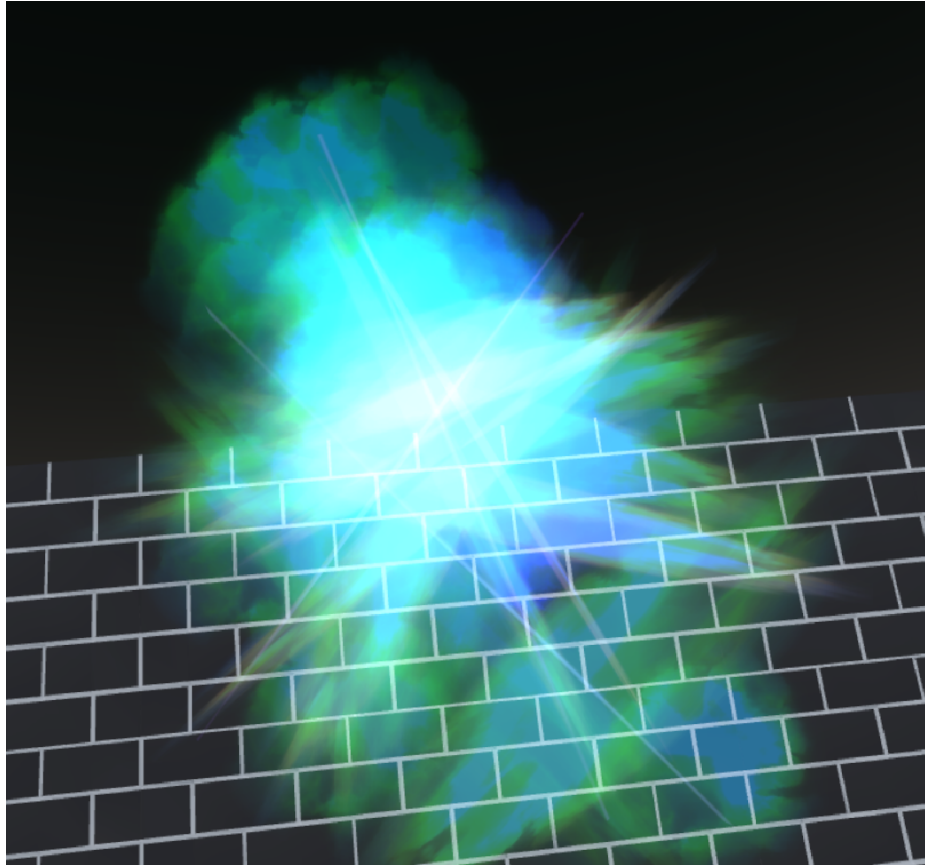


Figure 3.7: The fireball is used for the bomb explosion power.

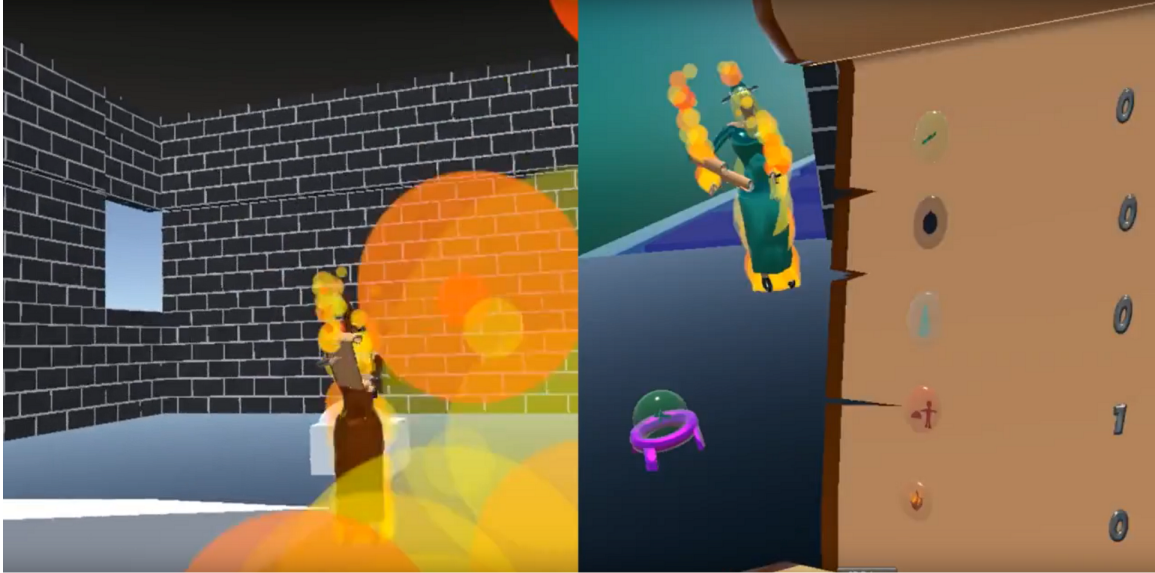


Figure 3.8: A split screen (two images from different player’s screens). When the fire power is selected by both players, it activates the fire particle system.

create a UI in a meaningful and useful way [20] for the user. When the swap power button is pressed, a function gets called within the powers script which then accesses a function that is located in a separate swap script. A similar method is used when selecting the bomb power, in which there is a separate script for the bomb power that is attached to the explosion prefab that is accessed in the powers script. The remaining powers are accessed and implemented in the powers script while having other scripts attached to the parent `GameObject` for networking purposes.

To get the fire and ice powers to work properly, the `OverlapSphere` physics function is used to detect when the hands of the witch are touching the ice cube or the water projectile. While a player has the ice power activated, they can freeze the water projectile. When the player has the fire power activated, they can melt ice cubes and cause them to disappear. For the fire power, it was also necessary to add an `OverlapBox` to melt the ice cubes when a collision is detected between the ice and the rest of the body. When the fire power is selected, a fire particle system is activated that engulfs the witch object in flames. When the ice power is selected, it changes the materials used for the witch’s skin color into a transparent light blue

material. The shrink power changes the local scale transform of the player prefab to a smaller size. The swap power moves the player prefab in a way that allows the witch model and camera to move to the position of the GameObject it is switching with. That GameObject then moves to the position of the witch model. The bomb power creates a custom prefab fireball object and when the fireball object detects a collision, it instantiates an explosion prefab.

Sounds had to be added to each of the powers. Royalty-free sounds or sounds we recorded were used for the powers. They are essentially open-source and can be used by anyone. The sound used for the swap power sounds like a slab of concrete being shifted across another hard surface. The sound for the shrink power sounds like rubber being stretched. The bomb power makes a loud bang when the fire ball collides with another object, using a royalty-free sound. The ice power is a custom sound made by crumpling a piece of paper and then editing the effects in an online music tool called Audacity [3]. The fire power uses a built-in sound in Unity that comes attached to the fire particle system. To attach each of the sound effects to the powers, a sound source component was attached to the witch and then specified in the powers script when the sound was supposed to be heard. The only sound that had to be specified differently was the bomb power, in which the fire ball spawns an explosion and the sound source is attached to the explosion.

3.2.4 Level Design

Demo Level: The first level was initially modeled in Blender, additions and edits were added afterward. The levels had to be designed according to the powers that would be used in that scene. There would have to be small constrained passages for the shrink power, platforms placed at higher locations that can only be reached by creating ice platforms from the octopus's bubbles, and empty spaces to place objects to either swap or blow out of the way with the explosion power. All these factors had to be taken into consideration when designing levels that would complement the usage of the powers.

The demo level features a puzzle that involves using the ice and fire powers to manipulate the position of a balloon. The objective is to use the ice power to make the balloon drop in height and the fire power to make it rise. The players repeat these actions until the balloon makes it out of a winding tunnel. Once that happens, the balloon rises above a platform. This is so a player can then use the swap power on the balloon to get to a higher location. The ice power was used to create a platform out of a water projectile to reach a swap power. After using the swap power to swap positions with the balloon to reach a high platform, a bomb power is then collected. The player on the high platform uses the bomb power to knock over crystal balls that contain the shrink power so that the other player can grab them. One of those balls is then passed to the other player so that both players can shrink and to reach the end of the level. The level can only be concluded once both players touch the square block at the end of the level. Upon doing so, a congratulatory message appears.

New Puzzle Level: As the demo level was made to illustrate how the powers could currently be used, another level was made to test the puzzle aspects of the game. The designing of this level involved the creation of several new models in Blender, which are purple barriers and buttons that are used to open them. The objective of this level is to figure out how to knock down crystal balls with the shrink power that are guarded by three barriers. A picture of the level can be seen in Fig. 3.9.

There are three buttons that correspond to the three barriers that are placed in separate ends of the map, while a lone cube sits in the middle of the level. While the buttons are pressed, their corresponding barriers are disabled. Two of the platforms require one player to use the other as a platform so that they can reach it. One player provides a hand for the other player to jump on to allow that player to reach these high platforms. There are also three swap powers and one bomb power available. The solution involves some set up. One player will need to bring the cube up to one of the platforms, while the other player will need to collect all swap powers that are within the level. One player will be called Player1 and the other Player2. Player2 will use Player1's help to reach a platform that is in front of a long highway filled with three

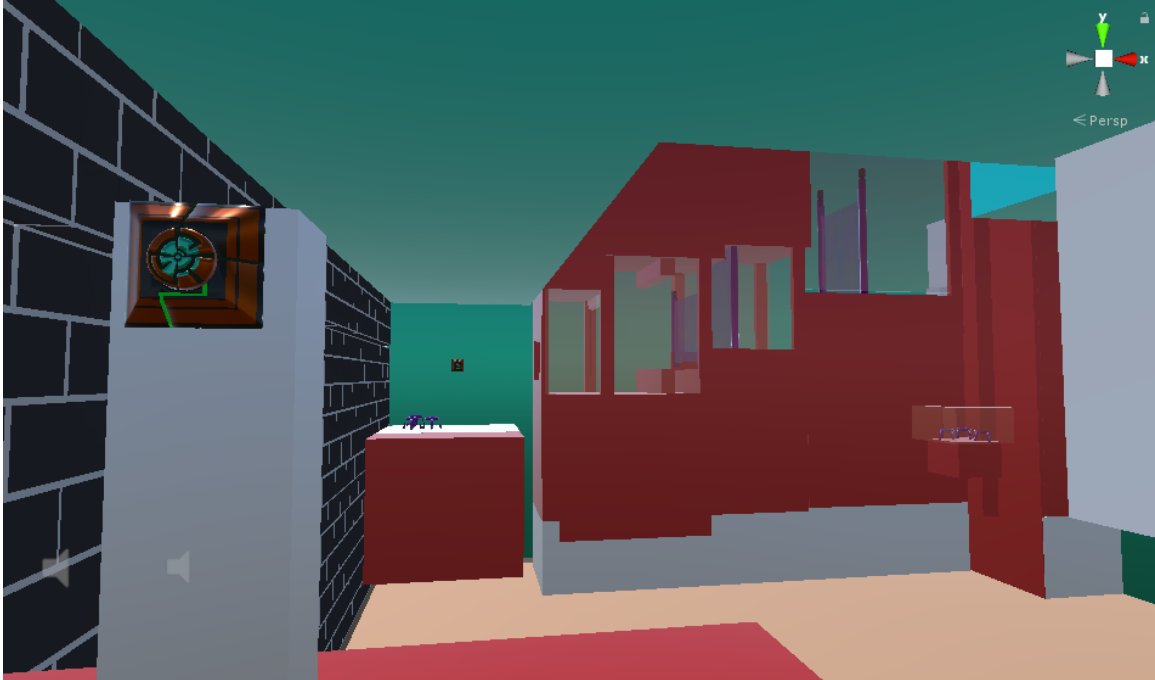


Figure 3.9: The puzzle level with the barriers, as well as the buttons the players use to open them.

barriers that ends with crystal balls that each contain shrink power. Player1 will go to the platform that does not have a cube. Player2 will slowly release a bomb spell towards the shrink powers. Since the bomb power is still active, Player2 cannot use other powers. Player1 will swap with Player2 so that Player2 can press the button located at the position Player1 is at. This opens up the first barrier. Player1 then swaps with the cube to press the button that is at that location. As the button is pressed, the next barrier is released. Once the bomb spell has passed that barrier, Player1 will swap with the cube again and then reach the final button as Player2 goes to collect the shrink powers as they fall down. The level is then completed.

3.2.5 Mirror Networking

Unfortunately, during the time this game was being developed the unity networking feature known as UNET was deprecated. The alternative used was the Mirror networking API found on the asset store or the Mirror public GitHub repository. There is a sub-branch of the Mirror API known as FizzySteamyMirror that allows the users

to link a host and client-server using their steam IDs [24]. Once FizzySteamyMirror was downloaded and installed successfully into Unity, the next hurdle to overcome was to sync up the player's movements between the server and client. To accomplish this, a networking transform child was added to the appropriate GameObjects of the player prefab, along with a script to disable any action that does not belong to the player on their side. After these tasks were accomplished, the player's arm movements and walking animations were visible on each others' screen.

After both of the character's movements were visible on both the server and client, it was time to make sure that the powers worked online. To achieve this, a networking script had to be added to the root GameObject of the player prefab for each of the five powers. These scripts are there to ensure that the game is synced over the network. After completing all five scripts, the powers used from either player could be seen by both users. Then Mirror's scripts were added to the appropriate power orbs so that the displacement of the power orbs, whether they are picked up or knocked out of place, as well as the position of the octopus, can be seen objectively on the same server.

To allow the players to see each others' arms move, the inverse kinematics scripts were kept enabled. They used the information about location of the hands and HMD sent from the other player to use for the inverse kinematics scripts to approximate the arm placement. For the bomb power, the local player had control over the spawned spell prefab. For the ice, the local player's materials are swapped and information is sent to the other player to change the materials of the non-local player. Something similar is done for the fire and shrink powers. When the client spawns a player prefab, the witch's materials for the clothes, hair, lips, and eyes, are changed so that the characters are distinguishable from the player spawned by the server. Networking eventually turned out to be a success, Fig. 3.6 and Fig. 3.2 show the two players interacting in different environments. There was other networking related work involving correctly spawning objects like the octopus, the balloon, and the spells and these are covered in detail in [24].

3.3 Gameplay

3.3.1 Locomotion

The locomotion method used included both room-scale and the controller to move. It was similar to glide locomotion [21]. Using the trigger by itself moves the player in the direction the player is looking in. Touching left or right on the touchpad, and then pressing the trigger would allow a side step. Touching back on the touchpad and then pressing the trigger would allow the player to move in the opposite direction that the player was looking in. The trigger was used since the touchpads seemed to jam easily. This locomotion method was chosen as it seemed simple to implement. A method such as teleportation was not used as it could look unusual to see a model repeatedly and instantaneously moving to different positions. Jumping and gliding were also added. When a player falls, they will fall at a constant, slow rate and can use the touchpad to move forward, left, right, or backward while gliding.

3.3.2 Positive Outcomes

The goals for the gameplay of this project included the possibility for depth in gameplay, the feeling of being on a team, variety in puzzles, and to make use of the motion controls that VR provides. One way that the game tries to encourage the feeling of being on a team is that players in the game can stand on each other. One player can hold out their hand to provide a platform for another player. This can be used as a method to separate players, or to make areas inaccessible without having to use powers. Another way the game tries to encourage a feeling of teamwork is the ability to pass collected powers to teammates. This is done by holding the model's hands to the button on the script and pressing the side buttons on the Vive controllers.

As mentioned before, the ice power was meant to encourage teamwork by allowing one player to freeze bubbles shot by the octopus while the other player tricks the octopus into sending them over. The fire and ice powers are not necessarily complete, as the original idea involved players not being able to enter certain areas unless those

powers were activated. For example, not allow a player not using the fire power to reach hot surfaces.

The balloons are meant to encourage teamwork by using the ice and fire power to cause the balloon to rise and fall. This step is repeated until it is in a position where a player can use the swap power on it. This idea was not explored greatly, but we believe it is usable for interesting puzzles. The bomb power takes advantage of the motion controls as it allows an explosion spell to be thrown. The swap power utilizes VR controls by using the HMD to aim. This power can use other players as objects to swap with, which encourages teamwork as it may be necessary to move a player to another location. Also, when a player is using a power, another power cannot be used. The other player would have to be in charge of using other powers which encourages players to choose roles.

Another important component added later in the game's development was the voice chat feature. Voice chat allowed the players to communicate with one another in the game while pressing and holding one of the touchpad buttons on the HTC Vive controller. This was done using mirror to send data over a network and using audio sources to play them [24]. The feature is a necessity since players will need to communicate their ideas to solve puzzles.

3.4 Conclusions and Future Work

3.4.1 Conclusions

This project demonstrates only a few of the countless exciting and innovative features programs like the Unity video game engine and Blender have to offer. However, the game was a successful project in the sense that it meets all the criteria initially set for it. It is a co-op game that not only reinforces teamwork but also requires it to make it through the demo. The five powers all have interesting visual effects and sounds attached to them, as well as situations where the players need to implement them. There is room for improvement in many areas of the game, but overall it is sufficient

for what it is intended. That being a great VR learning experience.

3.4.2 Future Work

While there are many items which could be added here, we will point out a few that we feel are important. Comfort mode [21], a method in which the user can turn their head without changing direction in the game could have been added for users who prefer it. Jumping, even with its potential to cause VR sickness [38], and the possibility of affecting immersion were kept in the game as it was deemed useful for gameplay purposes. A user study should be done to see how users feel about jumping and to gather feedback on the prototype. Since this game is still a prototype, the powers could be adjusted and more interact-able assets could be added. Other multiplayer services or libraries could be added such as Photon Unity Networking 2 [12], Dark Rift 2 [9], and Forge Networking Remastered [30]. All of which are available on the Unity Asset Store [32].

More levels could be added to test ways that the powers can be used and how they need to be adjusted. There also should be more GameObjects to interact with to help make puzzles more difficult and interesting. The original design of the scroll was intended to be dynamic and have buttons that represent powers placed on it in the order it was collected. This way, more than just five types of powers could be represented on the scroll, and the maximum amount of powers allowed to be collected by an individual player could be how many buttons could fit on the scroll. But to save time the scroll had all powers displayed next to a number.

Chapter 4

External Camera Integration: Merging Live Video Feeds for Remote Monitoring of a Mining Machine

This chapter is the draft of a publication which is in process of being submitted.

Abstract

This research entails using virtual reality to interpret video recordings in Unity from cameras on a machine used for mining excavations. The purpose of using a machine of this nature is to send it into hazardous mining environments rather than sending workers and having their lives jeopardized. This research is significant in the sense that it has the potential to show how two separate fields, such as virtual reality and robotics, can be combined to complete useful tasks. It also illustrates how machines can be used to replace workers in hazardous conditions in not only the field of mining, but other fields as well. Our contribution to this project will be to create a panorama of live video feeds captured by several webcams, and then make it so that the panorama of the live feeds can be seen in a VR headset. As we continue to develop this research, we hope to one day replace the webcams with cameras located on the mining machine.

Keywords: panoramic, virtual reality, robotics, mining, excavation, telecommunication, ITMM, hazardous, technology, software, operator, tasks

4.1 Introduction

This research entails using virtual reality to interpret video recordings in Unity from cameras on a machine used for mining excavations. The purpose of using a machine of this nature is to send it into hazardous mining environments rather than sending workers and having their lives jeopardized. This research is significant in the sense that it has the potential to show how two separate fields, such as virtual reality and robotics, can be combined to complete useful tasks. It also illustrates how machines can be used to replace workers in hazardous conditions in not only the field of mining, but other fields as well. Our contribution to this project will be to create a panorama of live video feeds captured by several webcams, and then make it so that the panorama of the live feeds can be seen in a VR headset. As we continue to develop this research, we hope to one day replace the webcams with cameras located on the mining machine.

For visualizing how the cameras should be placed on the excavator machine, we used a 3D model given to us by the mining department at the University of Nevada, Reno and imported it into Unity [13]. We have been contacted by the professor in the mining department in charge of the Immersive Teleoperation of Mining Machines (ITMM) project, and asked to create the software for another type of machine instead. The new machine we have been instructed to develop the software for is known as a rotary mining drill rig. The reason for having us switch machines is due to the time it will take for the mining department to finish constructing the excavator. The rotary mining drill however, is a machine that already exists, which will allow for the immediate implementation of the software once it has been developed.

The rest of this paper is structured as follows: The Concept is described in Section 4.2. Requirements is presented in Section 4.3. Use Case Modeling is outlined in Section 4.4. Design and Implementation are reviewed in Section 4.5. Merging Live Video Feeds is analyzed in Section 4.6. Related Work is discussed in Section 4.7, and Conclusions and Future Work are covered in Section 4.8.

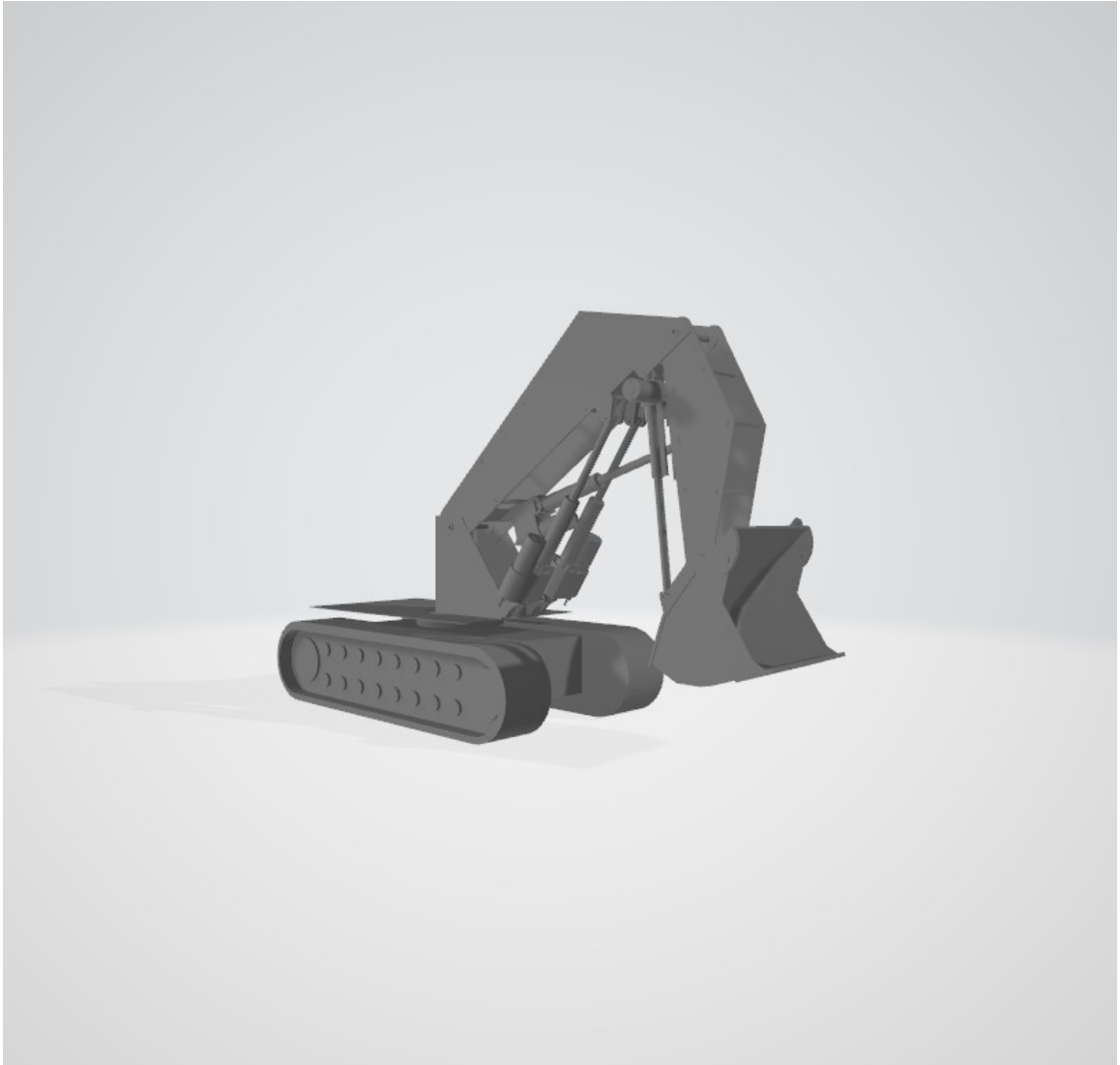


Figure 4.1: Model of the excavator mining machine provided by the mining department at University of Nevada, Reno.



Figure 4.2: Model for the mining drill rig, retrieved from [11].

4.2 Concept

The main goal of our contribution to the ITMM project is to merge live video feeds from cameras placed on a mining machine. Once the feeds can be viewed in a panoramic frame, the next step will be to develop software that will allow the use of a virtual reality headset to view the panoramic video of live footage taken from the machine. The tools we intend to use to develop this software are the Unity gaming engine [13] and Blender [4]. For future work, a more accurate model of the machine will be created in Blender and then tested in the Unity gaming engine to find the best camera placements. In order to program the software in Unity, it will require the use of the C # programming language. The library and language we use for the first method to create the panorama is OpenCV for Python, which will also be useful for stitching the video recordings together. The intended users of the mining machine will be typical mining workers at excavation sites. The goal is to make the software easy enough to use so that the technology can be utilized by anyone with minimal training.

The users will benefit by not having to go into hazardous mining conditions, such as a recently blasted site and instead send in the machine to complete tasks. It would also be helpful to create a virtual training environment for new operators to practice what it would be like to use the machine in real-life. The goal of ITMM is to eventually allow the users to operate the machine using a virtual reality headset along with a joy-stick controller. Not only does this technology have the potential to save lives in the field of mining, but it can also be used to accomplish other dangerous tasks such as repairing levies during hurricanes and deep sea exploration. This technology can also be useful for space exploration, in which the user can experience a more realistic view of stars and the surface of asteroids by seeing it in virtual reality instead of on a video recording terminal. It would also be an exhilarating experience to be able to operate a machine that is located on the surface of mars with the sensation that you are actually there performing the task the machine is.

4.2.1 The Problem

The problem that is being fixed through the implementation of this software is the necessity to send human workers into hazardous mining conditions. Shortly after blasting a hole in order to create a new mining tunnel, the conditions it creates for miners to move into those newly blasted holes are often unfavorable environments in regards to safety. The soil that makes up the floor and ceilings of that newly blasted tunnel can be unstable, and therefore the tunnel is prone to cave-ins of both the ceiling and the floor. In order to remedy this situation, a solution must be implemented that can replace the workers hired to move into these tunnels with something that can be subjected to the unsafe environments in their place, and can also maintain similar levels of visibility as an actual person. The proposal mentioned in the introduction above is a potential solution to this problem, and further details are discussed throughout the paper of how this technology can be used effectively.

4.2.2 Tools for Testing

The machine that the software will be developed for has not yet been fully assembled. Therefore, the software will be created and tested using the Unity gaming engine [13]. Once the model and the scene is prepared, camera objects will then be placed around the machine to get a better idea of where the cameras should be placed in real life. The current model of the machine can be seen in Fig. 4.1. OpenCV is a library written in C++, Python, and other languages. We will be using the Python version of the language to do the video stitching for one of the merging methods.

OpenCV and Python will not only come in handy merging the videos together for one of the methods, it will also be useful loading the resulting panorama onto a web-server using Flask. It may be a daunting task to smoothly merge the videos together from cameras that are elevated at different heights on the machine which is why the other Unity program we are creating to test the camera placement on the model will come in handy. Once the videos from the cameras are cleanly merged together and can be seen in a VR headset, it will mark the completion for stage one

of this research. We plan to complete more stages for this research in the future.

4.3 Requirements

4.3.1 Stakeholder Interviews

Stakeholder interviews were taken by two graduate students in the Mining Engineering Department at the University of Nevada, Reno in order to get a better idea of how the software should function. These students are also researchers in the ITMM project and are referred to as Interviewees 1 and 2 (I1 and I2):

Q1: What is the main function or purpose of the mining machine, what is it designed to do?

- I1 Answer: It is meant to load, haul, and dump all the muck shortly after blasting.
- I2 Answer: Use AI to collect data from operators, used to generate data through an automated system. It is also an excavator.

Q2: Who will be the primary operators of the machine, from which department (people with experience in VR)?

- I1 Answer: Equipment operators and minors, the goal is to have anyone regardless of educational background be able to operate the machine with minimal training.
- I2 Answer: The goal is to make it intuitive for anyone to use, no matter what educational background.

Q3: Do you think using virtual reality is the most efficient form of analyzing visual feedback from the machine, if not what is?

- I1 Answer: No, actually being there in person is. Virtual reality is not as accurate, however, it is still safer than being there in person.

- I2 Answer: In person is still the best, we are currently trying to make VR more efficient.

Q4 What is the expected number of cameras that will be placed on the machine and in what locations?

- I1 Answer: I'm not sure, more than four. Enough to cover 360 degrees with moving parts.
- I2 Answer: I don't know for sure, enough to capture a 360 degree surrounding.

Q5: When do you expect the actual model of the machine to be fully developed and ready to test the software?

- I1 Answer: Next year, summer. Possibly sooner.
- I2 Answer: We are still waiting for parts to arrive, hopefully I can finish it by January.

Q6: How will the movement of the machine be controlled, through what form of UI? (in the early stages)

- I1 Answer: Through a joy-stick, similar to how video games are played.
- I2 Answer: With a joy-stick controller, and eventually with AI.

Q7: Will it be possible to control the machine with the UI while wearing a VR headset, or will it need more than one operator?

- I1 Answer: Yes it will, maybe there will be a virtual representation of the UI in the virtual environment.
- I2 Answer: Yes it will, the same way you can play video games without having to constantly watch which buttons you push. Not entirely sure what the controller mechanism will be yet, the controller system is not finished.

Q8: What do you think about the use of rotating cameras on the machine and the possible software implementations that can be made with them?

- I1 Answer: It is do-able, but the rotation aspect might make it more likely for dirt to get caught in it.
- I2 Answer: A lot of dirt and dust may get into the camera. Already may be using a form of 360 degree cameras already anyways. Not sure if they are rotating ones though.

Q9: What other fields can you see this kind of technology of using virtual reality to control a machine from a remote location being useful in?

- I1 Answer: Exploration of tunnels, national tunnel systems, exploratory robots, NASA robots in space, other hazardous sites.
- I2 Answer: Aerospace, space travel, moon exploration, underwater construction and research.

Q10: What future work can you currently conceive being done on this project and other potential projects regarding the mining machine?

- I1 Answer: Sensors, real-time positions of every component in order to render a 3D model.
- I2 Answer: A goal is to automate the machine entirely without the use of an operator.

4.3.2 Software Requirements

In the following, the levels of requirements indicate their significance and priority and consequently the strictness of planning and completing their implementation. For example, all level 1 (highest priority) requirements were implemented and tested, while level 3 (lowest priority) requirements were left for future extensions of our software.

Level 1 functional requirements: Video camera objects should be added to virtual model of machine. Video recordings should be clipped together into one panoramic frame. User should be able to see the frame through a VR headset in Unity.

Potential challenges and limitations for level 1: Position of the video cameras on the machine may affect how video recordings are processed etc. one camera will be placed at a higher level than the others. Real-life machine may not function similarly to the machine model in Unity. Quality of the recordings may be affected by telecommunication. Unity may not be able to simulate real-life environment accurately.

Level 2 functional requirements: Locomotion should be attached to a joystick similar to the one mentioned in the interviews. Video recordings should be rendered more clearly based on which direction the user is facing by having the cameras on the machine rotate in that direction.

Potential challenges and limitations for level 2: The behavior of the rotating cameras in real-life might not be similar to the functionality of those available in OpenCV. Locomotion on the actual machine may experience a lag effect due to telecommunication, no way to know for sure until machine has been fully developed.

Level 3 functional requirements: The rotating cameras should change position through the use of eye-tracking.

Potential challenges and limitations for level 3: Eye tracking may be too new of a technology to implement for the use of this kind of system i.e. VR telecommunications AND operation of a machine.

4.4 Use Case Modeling

A use case diagram of the software is shown in Fig. 4.3.

Initiate Machine: This process is activated when the user initiates the software of the machine by powering on the VR headset. All system components are then activated such as the cameras and then the machine awaits further instruction.

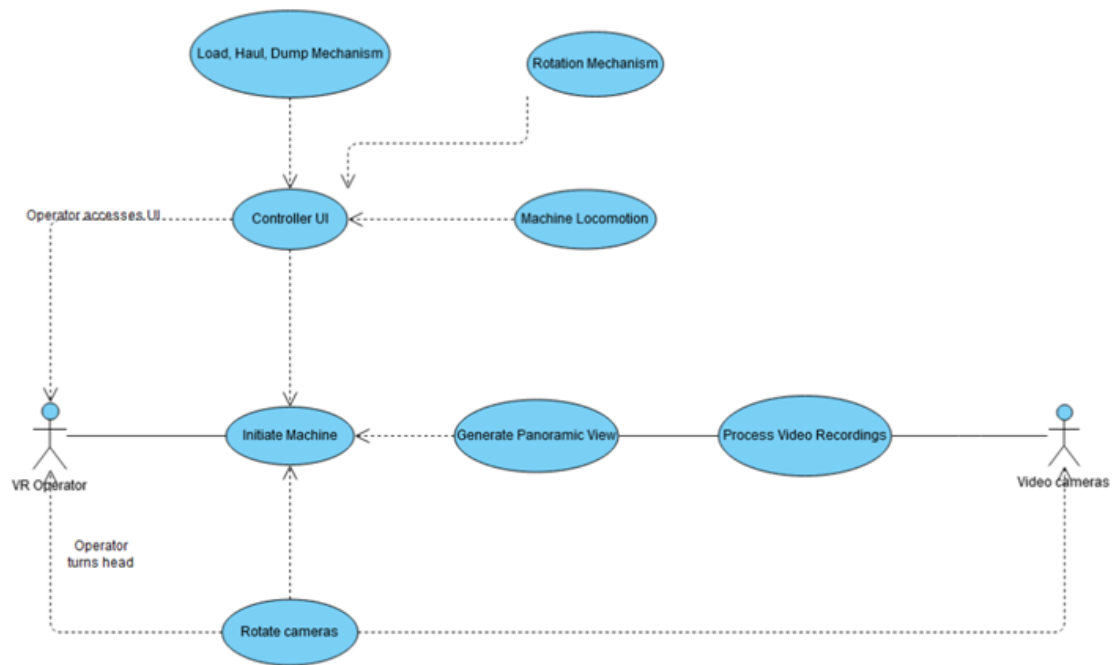


Figure 4.3: This image displays the use case diagram of the proposed software.

Process Video Recordings: This process is activated once the machine has been initialized and the camera actor begins recording information. The recordings are then processed which then activates the panoramic view process.

Generate Panoramic View: This process is activated once the video cameras begin recording and the recordings are then processed. Once the recordings have been processed, the panoramic view process generates the recordings into one 360 degree frame. This chain of process is dependent on whether or not the machine has been initiated.

Rotate Cameras: This process is only activated if the operator wearing the VR headset turns his/her head, the machine has been initiated, and the video cameras are operational. If all of these parameters are met, then the cameras will rotate their position.

Controller UI: This process is activated if the operator accesses the UI, and the machine has been initiated. If both of these parameters are met, then the machine

awaits further instruction from the UI. **Machine Locomotion:** Once the controller UI process has been activated, the machine locomotion is then an operation accessible through the UI. This process governs the overall horizontal movement of the machine based on the inputs to the UI.

Rotation Mechanism: This process governs the rotation of the machine about the vertical axis. It is useful for rotating the machine when it is stationary. This process can only be accessed if the controller UI process has been activated.

Load, Haul, Dump Mechanism: This process governs the movement of the bucket and crane attached to the machine. This mechanism is used to load the bucket with debris or dirt, haul it to another location through the use of the locomotion process, and to dump the debris or dirt. It is only accessed if the controller UI process is activated.

4.5 Design and Implementation

4.5.1 Block Diagram

To get a better idea of how the design of the software should look, a block diagram was constructed and is displayed in Fig. 4.4.

4.5.2 Design Class Diagram

A class diagram of the software is shown in Fig. 4.5.

VR: This class records the direction in which the operator is looking, and then applies the images gathered by the cameras to correspond to that direction. It acquires the direction through the get methods present in the class.

Cameras: This class uses four camera objects to take video recordings via the recordVideo method.

Stitch Image: This class takes the four recordings from the camera in the form of images, then stitches those images together into a panoramic frame with the imageStitch method.

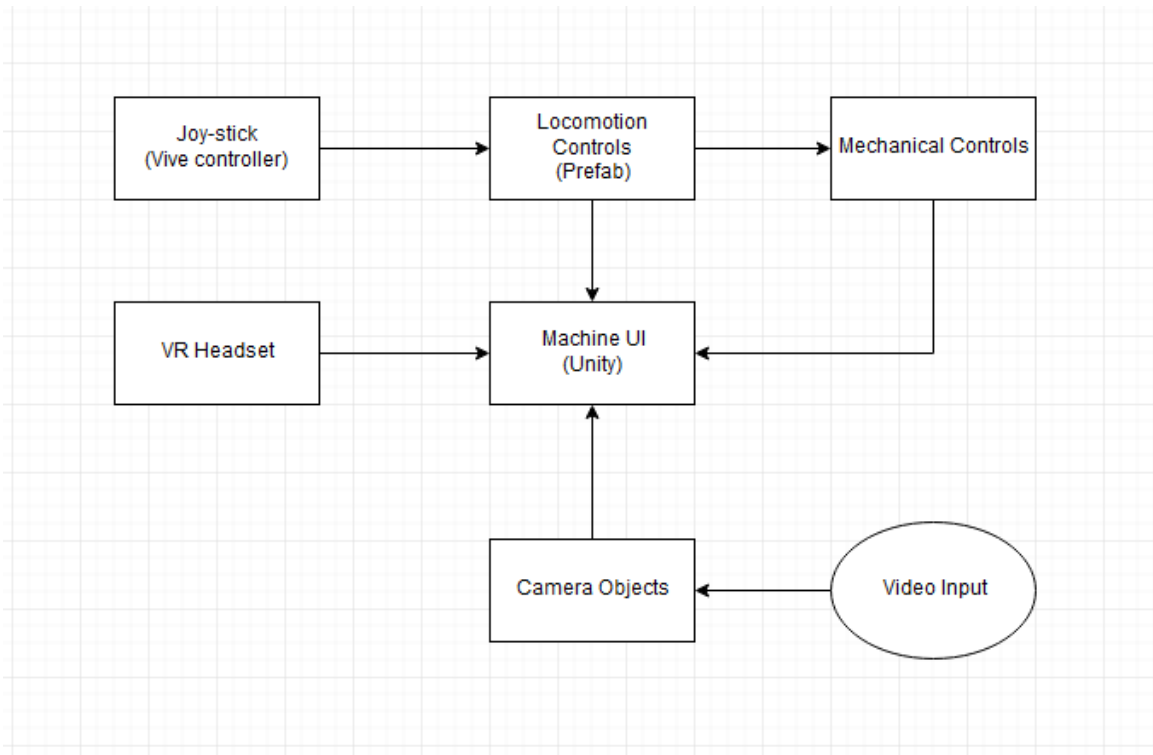


Figure 4.4: This image displays a block diagram of how the system is supposed to function. The operator uses a VR headset and HTC-Vive controller to control the machine. The VR headset connects directly to the UI, whereas the Joys-stick connects to the locomotion prefab, which can alternatively control the mechanical aspects of the machine. The camera objects read in video input and send it to the machine UI.

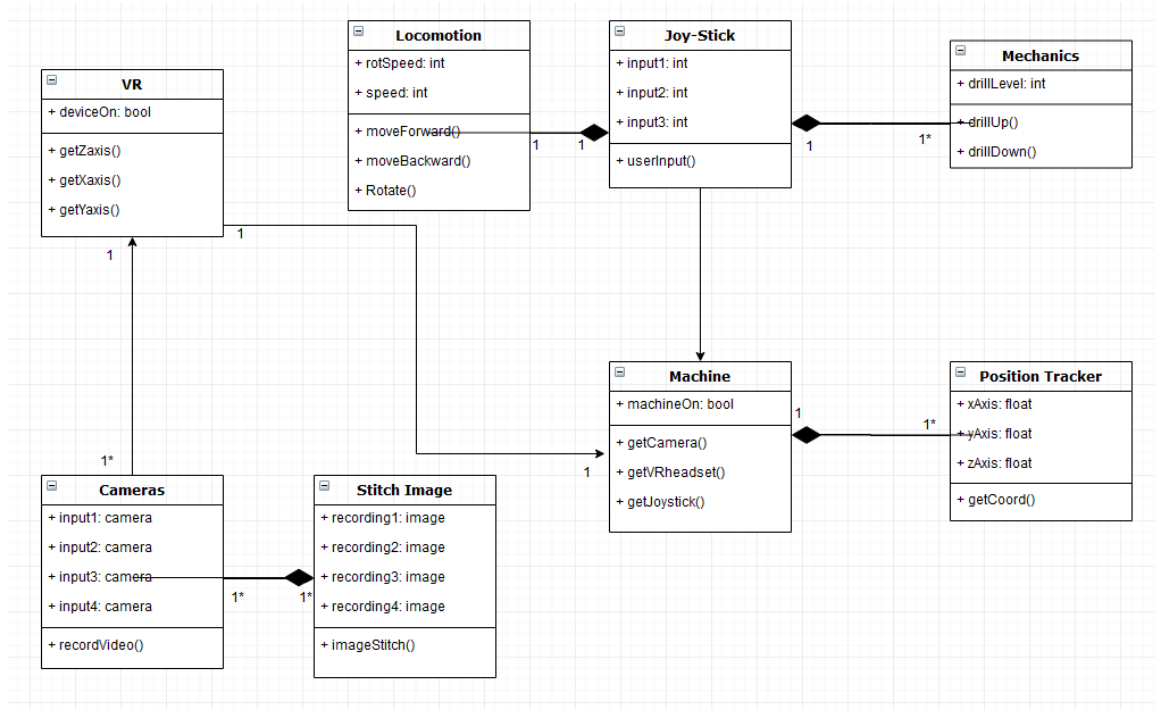


Figure 4.5: This picture shows the design class diagram of how the software will be constructed.

Joy-Stick: This class records three input values in the form of integers, whichever input is chosen depends on the userInput method.

Locomotion: This class is determined by the type of input selected from the Joy-Stick class. The input from the Joy-Stick determines whether the moveForward, moveBackward, or Rotate method is called. The speed and rotSpeed are recorded as integer values in case one wishes to increase the speed of the locomotion.

Mechanics: This class governs the up and down motion of the drill animation attached to the machine. An integer value records the current height or level of the drill, whereas the drillUp and drillDown methods control the actual movement of the drill.

Position Tracker: This class determines the current location of the machine, this feature will come in useful when the actual model of the machine is out in the field. Three float values record the x, y, and z coordinates of the machine. The getCoord method returns the exact coordinates of the machine by combining the

three float values.

Machine: This class is the central interface for all of the other classes. If the bool value of the machine is true, then the getCamera, getVRheadset, and getJoystick methods are then activated. This class governs the control of the machine model in Unity.

4.5.3 Data Design

The processes that make up the software will run in a manner in which a process will be activated one after the other. Once the processes involved with locomotion are reached, they should be the last and most consistent processes that are continually executed. In order to best execute this behavior, it seems that a stack data structure would be the most suitable choice for this software. All other activities will be pushed onto the stack, and once the locomotion activities are reached, they will continually be popped on and off.

4.6 Merging Live Video Feeds and Testing Camera Placement

4.6.1 Using OpenCV for Python to Merge Webcam Footage

For merging the live video feeds of four webcams into a panoramic frame, two methods were used and analyzed to choose which one would be best in terms of visual quality. The first method consists of using openCV for Python and accessing the four webcams by USB port in a Python script. The method for merging the video feeds together using Python relies on information and code provided by [25], which utilizes a data structure to detect the homography matrix between two images. Once the homography matrix between two images is established, the two image frames along with the homography are then passed to the openCV warp perspective function to create a combined image from the two separate ones. The code is structured in such a way that the homography is not computed between video frames, but single images

that are run through a while loop so that the frames are constantly updated.

Once a panorama comprising of video feeds from four separate webcams was established, we then needed to find a way to get the resulting panoramic frame into Unity. We came up with the idea to load the resulting panoramic frame to a web-page, and then from that web page into Unity. A method for loading a single webcam frame to a web page using a Python script along with the web framework Flask is provided by [25], so combining the scripts for merging the webcam frames and the flask web server script gave us the result we were hoping for. The most crucial part of the Python script that merges the video frames and prepares them to be output to a web-page is shown in Fig: 4.6. After specifying the proper IP we wanted to send the panoramic frame to so that it could be accessed by other machines on the same network, gives us the following result shown in Fig. 4.7. The result is a panorama of live video feed from four separate webcams, which is not as clear of a picture as it could be. This is because different webcam models were used to capture the live feed.

Because the resulting panoramic video frame is technically a JPEG image due to the while loop described earlier, the Unity video player feature is unable to load it using the URL method. After doing a lot of research on what the most efficient way to load the panorama into Unity would be, we discovered that sectioning off and live streaming the webpage to Vimeo, and then using the Vimeo Unity SDK would allow us to load the page into Unity and then use the video stream as the background for a skybox. Unfortunately this method requires a Vimeo pro subscription which happens to be exceedingly expensive during the time of writing this paper. Another possible option would be to use the Embedded Browser Unity asset, but again that would require spending a decent amount of money. In order to analyze what the result would look like if we could successfully load a live panoramic feed into Unity, we decided to take a video recording of the webpage the same way Vimeo would do, and we used the recorded video clip as the background for the skybox. The result of this is displayed in Fig. 4.8. The result displays a live video feed of someone walking through a room, however, the resolution is of low quality.


```

@app.route("/")
def index():
    # return the rendered template
    return render_template("index.html")

def display_webcam():
    # grab global references to the video stream, output frame, and
    # lock variables
    global A_stream, B_stream, C_stream, D_stream, merger, outputFrame, lock
    # initialize the motion detector and the total number of frames
    # read thus far
    # loop over video stream frames
    while True:

        #retrieve frames from their corresponding video streams
        A = A_stream.read()
        B = B_stream.read()
        C = C_stream.read()
        D = D_stream.read()

        #resize the frames
        A = imutils.resize(A, width=400)
        B = imutils.resize(B, width=400)
        C = imutils.resize(C, width=400)
        D = imutils.resize(D, width=400)

        AB_stream = merger.mergeStream([B,A])

        CD_stream = merger.mergeStream([C,D])

        frame = merger.mergeStream([AB_stream, CD_stream])

        with lock:
            outputFrame = frame.copy()

```

Figure 4.6: Each of the four frames are merged together and stored in a single variable, which is then used as the output frame.

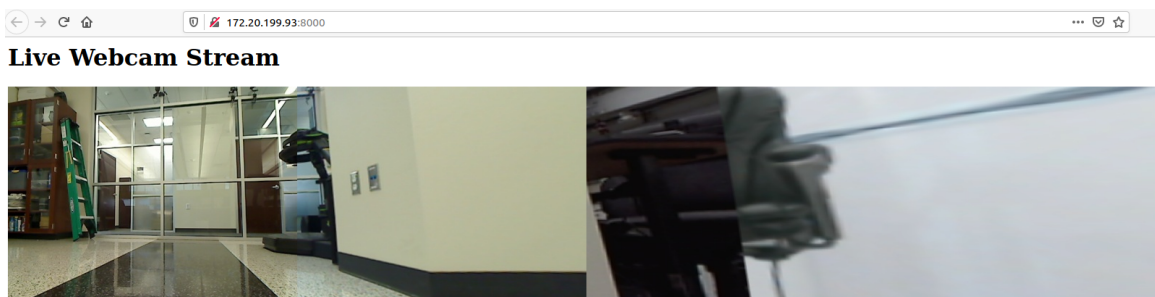


Figure 4.7: This image shows the result of merging four webcams together and then loading that resulting frame to a web server.

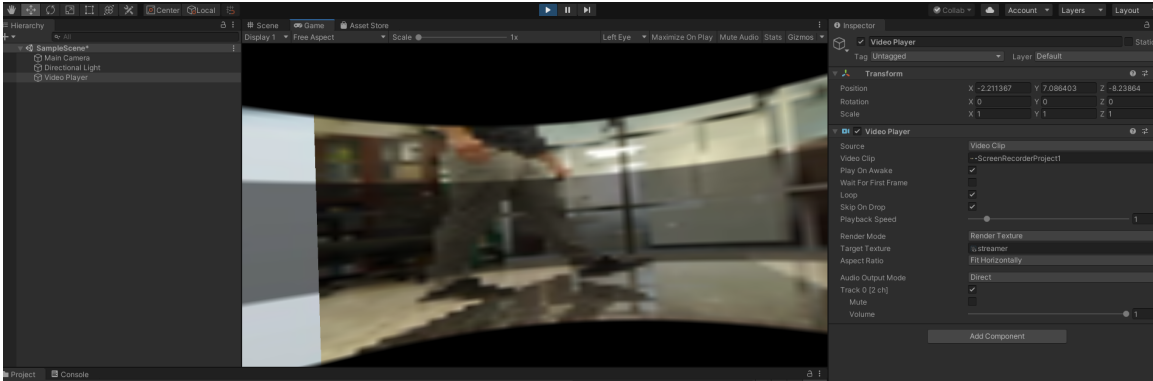


Figure 4.8: The result of recording a video clip of the panoramic web page and using it as the skybox background in Unity.

4.6.2 Loading Webcam Footage Directly to Unity

The second method for merging four webcam frames is far less complex, and ironically, yields a far greater result in terms of resolution. This method entails using a C# script to identify the webcams that are connected to the computer, and then loads the live feed from the webcams directly into Unity. The script is attached to whatever object the live feed will be displayed on, and it uses the render texture feature to accomplish this. The script automates the whole process of creating and attaching a render texture to a polygon and can be viewed in Fig 4.9, after which it is up to the programmer to place the polygons next to each other in a careful manner to get a panoramic image from the four cameras. The result of this method is shown in Fig. 4.10, and as you may find, the resolution is of far greater quality than the previous method.

4.6.3 Testing Placement of Cameras on the Machine Using Unity

In order to establish the best positions for placing the cameras on the machine to obtain a clear 360 degree recording, a model of the mining drill rig machine was loaded into Unity and camera objects were placed in strategic locations on the machine. To get a clear perspective of what the cameras were capturing, render textures corre-

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class testScript : MonoBehaviour
{
    void Start()
    {
        WebCamDevice[] devices = WebCamTexture.devices;
        Debug.Log("Number of web cams connected: " + devices.Length);
        Renderer rend = this.GetComponentInChildren<Renderer>();

        WebCamTexture mycam = new WebCamTexture();
        string camName = devices[1].name;
        Debug.Log("The webcam name is " + camName);
        mycam.deviceName = camName;
        rend.material.mainTexture = mycam;

        mycam.Play();
    }
}
```

Figure 4.9: The C # script used to load in a live webcam feed directly into Unity and render it to a polygon.

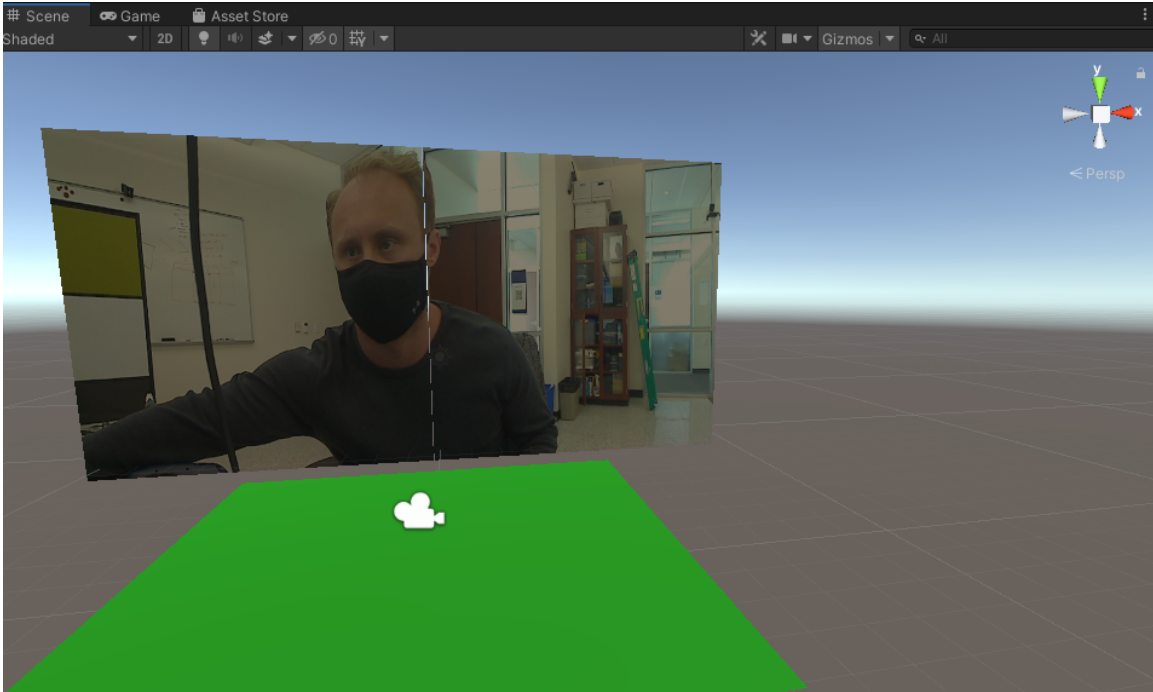


Figure 4.10: The result of loading the webcam feeds directly into unity and attaching them to polygons in order to create a panorama.

sponding to each of the camera objects' footage were attached to 3D planes in such a way that shows an outline of a panorama. This is shown in Fig. 4.11. We took this a step further and applied each of the six camera objects' render textures to a six sided skybox to gauge what it would be like to view the camera footage from the perspective of an individual wearing a VR headset viewing the merged frames. This is displayed in Fig. 4.12.

4.7 Related Work

4.7.1 VR and Robotics

In our search to find projects related to using VR to operate machines, we came across a conference paper [22] that describes a method that uses VR to train operators to control waste management robots. To train these operators, the participants are placed in a VR simulation that mimics how controlling the robots in real-life would be. To do this, the operators use voice commands and hand gestures in order to

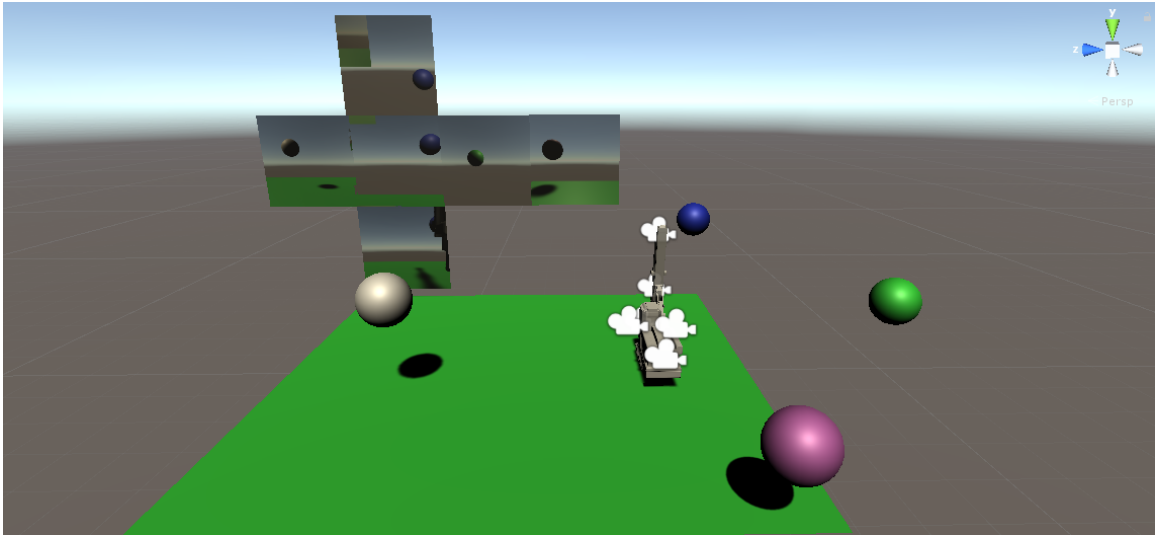


Figure 4.11: This is an image captured of the scene view in Unity of the prototype. Here you can see the mining machine model with the camera objects attached to it, along with the render textures set up in an order to create a panoramic frame.

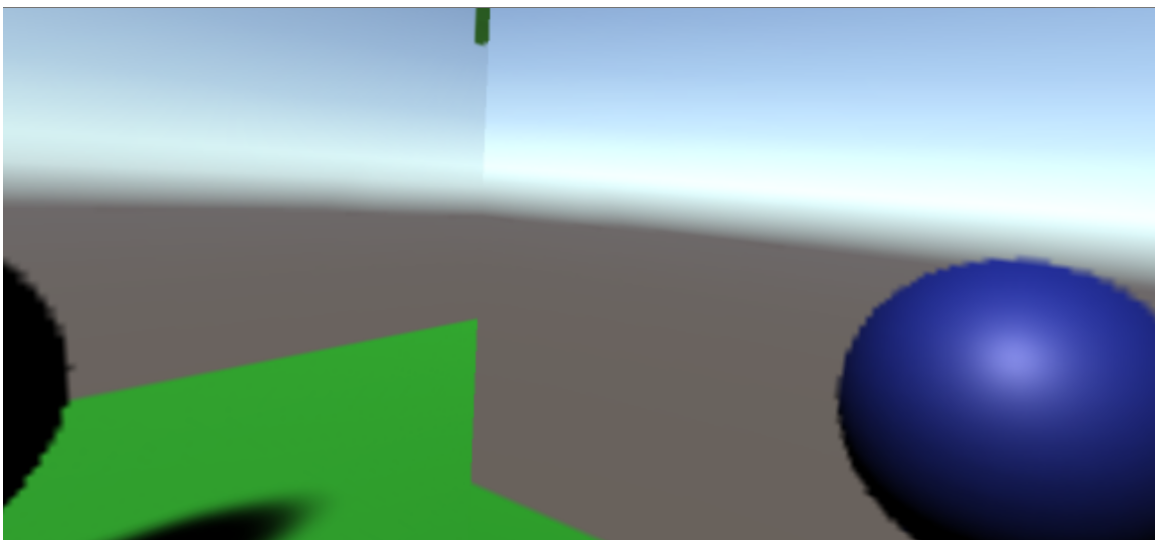


Figure 4.12: Here is a snapshot from the game view window in Unity of the prototype. This shows the viewer within the skybox, which is meant to display what the user sees out of the camera objects placed on the machine.

create a realistic and natural experience. The robots that the operators are being trained to use are for cleaning up trash in toxic environments. At the end of the paper, the authors state that in their future work they intend to allow the robots to be controlled using VR as well. The statement in the last sentence is what makes this paper incredibly relevant to what we intend to design for our research.

Another exceedingly useful paper that we discovered was one that describes three areas where VR and robotics can be successfully integrated to solve certain tasks [5]. According to the paper, robotics and newer VR research communities have been working in isolation. Both of these fields can benefit greatly when combined together. The first area where an integration of the two fields would compliment one another is VR-enhanced CAD design, plant layout simulation, and robot programming. A subsequent area that would also see huge benefits if the two technologies were combined is how VR can be used in supervisory teleoperation, for collaborative control of a single robot by multiple operators, and for single operator-single robot systems. Finally, the paper discusses how Robotics can benefit VR in general pertaining to the ability of robots to serve as force feedback interfaces to a VR simulation.

4.7.2 Relevant Projects and Applications

A relevant project that pertains to this field of study is the work being done on the Spherical Fish Tank Virtual Reality Display(SFTVR) [40]. What makes this project relevant is its focus on displaying 3D spatial effects by using head-tracking to render an image to the perspective of the viewer. It combines Fish Tank Virtual Reality with a spherical display, which can have promising effects on increasing the user's performance for 3D tasks. It is clear the advances made as a result of optimizing such a project being useful for future work in our own project. One such application as a result of the research done on the SFTVR could be designing a UI [19] to control the mining machine within the virtual environment. This stage of the mining machine project would not come until later, for now the primary focus is to render clear images to be seen through a VR head-set.

Another useful application worth mentioning is the use of visual manipulation and kinematic data collection in VR [27]. This journal article aims to validate the HTC VIVE VR system for its ability to effectively collect kinematic data by assessing the accuracy of its orientation signals and position. The evaluation was conducted by taking the VIVE controller, as well as the tracker, and comparing them to a Polhemus Liberty magnetic tracking system sensor. The results of this evaluation concluded that the HTC VIVE system can be used as a reliable means of collecting kinematic data. This information again comes in handy pertaining to future applications in our research involving creating a UI in the virtual environment. Perhaps the HTC VIVE [14] controllers can be used for tracking the user's hands on the 3D UI, or even be used to control the machine without the need of another UI.

4.7.3 Other Related Areas

There are a host of other resources pertaining to this field of research in the form of other major VR research labs, as well as patents for new devices that can aid in future work for my project. One of these major labs includes the NYC Media Lab [16], which is currently working on 14 new exciting projects. These projects not only enhance the way people experience entertainment, some of them also have the potential to help workers of multiple different fields. The project Acculis for example, is being developed for the construction industry to facilitate the transfer of information between designers and construction workers.

Another VR lab that appears to be conducting extensive research in the subject is the CITL'S Virtual Reality Lab [10] located in Illinois. The purpose of this lab is to serve as a research and testing grounds for students and faculty who share an interest in exploring the educational potential VR has to offer. They are multiple different headsets available at this lab including the HTC VIVE Pro, the Oculus Rift, as well as the Oculus GO. The innovation studio located in this lab is designed to observe the learning strategies and behaviors of individuals using the devices in order to use them as mediums to further explore educational content.

One of the patents we located on google [34] describes a 360-degree virtual reality camera system. This device captures a 360 degree image of a local area, in which a portion of it is in stereo. The camera system includes a first rigid plate, a plurality of axis cameras, and a second rigid plate. The peripheral cameras are aligned in a ring configuration which can be useful for allowing objects past the image threshold to be within fields of view of at least two of the cameras. Another google patent [36] discusses the use of data structures and delivery methods that can expedite VR playback. This can be used for both virtual and augmented reality. The concept is that a video stream or augmented reality experience can be stored and then delivered to the user. The patent discusses how the stream is divided into plurality of units based on viewpoint segmentation, time segmentation, as well as orientation segmentation.

This google patent [35] discusses a new method for remote virtual reality control of move-able vehicles. This method includes displaying a graphic model of a portion of the vehicle on a remote output device. The vehicle also has one or more move-able partitions. This method involves image processing from one or more devices, and can identify one or more vehicle commands. The last google patent [33] that is relevant to our research topic outlines the idea of a three-dimensional scene reconstruction method that uses a set of two dimensional images. The purpose of this is to enable better preservation and sharing of immersive experiences. The set of images that are used are taken from different vantage points, and can process each image to extract depth information and then stitch the images in a multilayered panorama.

This journal article [7] portrays the capabilities of current, low-cost VR technology. It discusses recent advancements in devices such as the HTC VIVE, the Leap Motion Controller, the Oculus Rift, and the Microsoft Kinect. Each of these devices brings an immersive VR experience into the homes of consumers at a much lower cost and special requirements than in recent years. These devices are now lowering the barrier for the entry of VR engineering applications. Another relevant paper [26] outlines three different input technologies which are vision-based motion controls, gamepads, and the room scale feature, for an interactive VR environment. The sys-

tem is able to visualize data from several sources by enabling the user to browse and interact with displayed information in an immersive VR environment. This paper implements a user study to show the experienced workload, as well as perceived flow of interaction.

4.8 Conclusion and Future Work

In order to formulate a clear picture of the type of software we will be designing, extensive research into VR and exploration had to be conducted. Our objective was to create a realistic representation of how the software will function in Unity. Useful tools for testing the software include Unity, OpenCV, and the Flask web framework. There is already research being done that involves the integration of both VR and robotics, but the unification of the two fields still appears to be in its nascent form compared to what it can be. There are a host of other resources related to VR and exploration in the form of major research labs, projects and applications, and patents found on google.

Once we were able to successfully clip the camera images together into a panoramic frame, and then allow that frame to be viewed in an HTC VIVE headset in Unity, it marked the completion of stage one of this research. There are many other stages for the development of the research we intend to participate in, such as the integration of a joystick controller to operate the machine. Another goal could be to create and implement a 3D UI within the virtual environment that mimics the actual controls of the joystick controller. This 3D UI will allow the user to see which buttons they are pressing on the joystick controller in the virtual environment so they do not have to remove the headset for any reason. Another possible project that we may take part in involves implementing the 360 degree cameras mentioned earlier to then aim the cameras in whichever direction the user is looking via eye-tracking technology. This type of effect can give the user more control over what they see, rather than being at the mercy of four or five stationary cameras.

Some other future work to consider could be to integrate light detection and

ranging (LIDAR) sensors to the mining machine to either complement, or replace the cameras. These sensors would use light in the form of a pulsed laser to map the terrain of the environment the machine is present in, and then procedurally generate a graphical representation of that terrain inside of Unity. A 3D model of the vehicle could then be added to the scene with the generated terrain to simulate the position of the real machine with respect to its location in the actual terrain. Virtual Reality could then be enabled in Unity to give the user a view of the procedurally generated terrain from the perspective of the machine model.

When implementing the use of a joystick controller to direct the locomotion of the machine, we could also allow the controller to manage the load, haul, and dump mechanisms. It would be a convenient feature to link the mechanisms of the joystick controller to the graphical model in Unity, so that whatever operations the machine performs as a result of the joystick controller in the real world will also be simulated in the Unity rendition. To achieve this effect, the joystick device that controls the machine should also control the model in Unity, in other words, the two control mechanisms should be in sync. To assign animations to the 3D model of the machine in Unity to simulate the load, haul, and dump mechanisms, the Unity add-on called Rigify can be used to attach rig controls and bones to the machine model.

There appears to be two foreseeable options for integrating remote cameras on the mining machine to use for the video merging methods. One option could be to use wireless cameras that are connected to a network, and then to use a tool such as Wireshark to identify the camera's IP. It should then be an option to specify in the video merging code to identify the cameras by IP rather than USB adapter. The other option is to continue using USB cameras, but to use the wireless blue-tooth variation and connect to the computer that way. The challenge is to find USB blue-tooth adapters that have a long enough range to reach the computer without there being too much path loss, also known as the deterioration of the signal. The best processing power for the computer that receives these blue-tooth signals could be anywhere from 2.3 GHz or higher. The main factor to getting the best video resolution is likely to

increase the strength of the blue-tooth signal and not the processing power of the computer.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Two forms of locomotion were analyzed and compared in a user study, one being a natural and the other an inorganic method. The natural method consisted of two sensors that were attached to the participant's feet, so that lifting their feet in an up and down motion would induce movement in the virtual environment. To induce movement using the inorganic method, the user would simply press down on the touchpad of the HTC-Vive wand. The user study concluded that the inorganic technique was not only the more preferred method of the participants, it was also more efficient in terms of the overall completion times. However, most of the participants agreed that the natural method was more immersive than the inorganic method.

We succeeded in developing a multi-user cooperative VR game using the Unity game engine and the Blender modeling tool. The game placed two players in the roles of witches equipped with scrolls that were used to access their powers. Powers were obtained by collecting spherical objects, or power orbs, that were placed in strategic locations around the map that would require a methodical and cooperative approach from both players to reach them. Once powers were collected, the players would have to use them to solve puzzles in order to complete the challenges. Again, the challenges were structured in a way that teamwork was essential to completing the levels.

We developed two methods for merging live webcam streams into a panorama for the ITMM project, which could then be viewed on an HTC-Vive headset in the

Unity game engine [13]. The first method consists of using OpenCV for Python to merge live webcam feeds together into a panoramic frame, along with the Flask web framework to load the panorama to a web page where it can then be imported into Unity. Unfortunately due to time and money constraints, we could not get this method to fully function. Instead, we chose to take a video clip of the merged panorama to load into Unity and apply it as a skybox background. The second method consisted of loading the webcam footage directly into Unity using a C # script and applying the live-feed as a render texture to a polygon. The panorama was then created manually by placing the polygons next to each other to connect the frames of each live-feed.

5.2 Future Work

There are plenty of aspects where each of these research areas can be improved upon, or further developed so that other user studies may be implemented.

5.2.1 Testing Other Forms of Locomotion

The user study featuring the natural and inorganic forms of locomotion showed that the inorganic method was preferred by most of the participants over the natural, or walking in place (WIP) method. This outcome could have been due to the design of the level, and perhaps the level design can actually have a significant impact over how the users perceive the locomotion technique. Maybe a game that is geared more towards solving puzzles, and not walking in a straight line to complete a course, would be a better option to use a WIP method of locomotion. There are also other forms of locomotion that were not tested in this user study, such as teleportation. This is when the user predetermines where he or she wants to travel in the VE, and with a single press of a button, will instantly appear in the spot they were targeting. This can be done in other ways than just pressing a button, such as the user lifting an appendage where a tracker is attached, blinking, or simply looking in the direction they want to travel. A future user study can be conducted with these other forms of

locomotion in mind and with other types of levels.

5.2.2 Improving the Multi-user Cooperative VR Game

One way the multi-user cooperative VR game can be improved is to spend more time adding more detail to the models. This can be done by adding more polygons to use in designing the facial features of the witches, allowing dynamic movement of their clothing so that the clothes do not seem so rigidly attached to their bodies, and using more polygons in general to add more detail to all of the models. In terms of game play, other levels can be created to allow the players to further utilize their powers in creative ways to solve challenges. These levels can also have more interact-able game objects, like the buttons that were used to lift the barriers. It can potentially add to the enjoyment of the experience if a comfort mode were added to the game as well. This would allow users to adjust in the settings an option to not move in the direction they are turning their head.

5.2.3 Merging Live-streams for Remote Cameras

The end goal of the webcam live-stream merging research is to apply the most effective video merging method to remote cameras placed on a mining machine, so that a single panoramic frame can be viewed in the Unity gaming engine with an HTC-Vive headset. We currently have the code structured to merge live webcam feeds together that are connected to a computer via USB cords. Applying this method to remote cameras could potentially entail making changes only to the part of the code where the webcams ports are identified. If the remote cameras are connected to a network, and a tool such as Wireshark is used to Identify the IP addresses of the cameras, then the code can theoretically be structured to identify the cameras via IP address instead of USB ports.

5.2.4 Implementing a Joystick Controller to Mining Machine Project

Another attribute we hope to add on top of the VR component to the mining machine project is implementing a joystick controller to control the locomotion of the machine. We hope that by implementing this feature, we may simplify what would normally require a sophisticated skill-set so that anyone with minimal training can easily learn how to operate this device. To allow the user to see the joystick controller while they have the HTC-Vive headset on, a 3D model of the controller can be created in the Virtual Environment that correlates to its location, as well as what buttons are being pressed, in the real world. This feature already exists when using the HTC-Vive wands, therefore it should not be too difficult to recreate this feature using a different model controller.

References

- [1] M. Al Zayer, P. MacNeilage, and E. Folmer. Virtual locomotion: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2315–2334, 2020. DOI: 10.1109/TVCG.2018.2887379.
- [2] Kurt Andersen, Lucas Calabrese, Andrew Flangas, Sergiu Dascalu, and Frederick C. Harris, Jr. A comparison between a natural and an inorganic locomotion technique. In Shahram Latifi, editor, *17th International Conference on Information Technology–New Generations (ITNG 2020)*, pages 317–323, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-43020-7.
- [3] Audacity. *Audacity* [®] — Free, open source, cross-platform audio software for multi-track recording and editing. (Last Accessed: April 16, 2020). Audacity. 2020. URL: <http://audacity.sourceforge.net/>.
- [4] Blender Online Community. *Blender - a 3D modelling and rendering package*. (Last Accessed: January 7, 2020). Blender Foundation. Blender Institute, Amsterdam, 2019. URL: <http://www.blender.org>.
- [5] G. C. Burdea. Invited review: the synergy between virtual reality and robotics. *IEEE Transactions on Robotics and Automation*, 15(3):400–410, June 1999. DOI: 10.1109/70.768174.
- [6] Lucas Calabrese, Andrew Flangas, and Frederick C. Harris, Jr. Multi-user vr cooperative puzzle game. In Shahram Latifi, editor, *17th International Conference on Information Technology–New Generations (ITNG 2020)*, pages 293–299, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-43020-7.
- [7] Joshua Q. Coburn, Ian Freeman, and John L. Salmon. A review of the capabilities of current low-cost virtual reality technology and its potential to enhance the design process. *Journal of Computing and Information Science in Engineering*, 17(3), 2017. DOI: 10.1115/1.4036921.
- [8] Valve Corporation. SteamVR plugin, 2018. URL: <https://assetstore.unity.com/packages/templates/systems/steamvr-plugin-32647>. (Last Accessed: October 10, 2019).
- [9] DarkRift. Darkrift networking. URL: <https://darkriftnetworking.com/DarkRift2/Docs/2.3.1/html/944c4100-5c17-449f-8a8e-c9fbfdaedae.htm>. (Last Accessed: December 30, 2019).

- [10] Center for Innovation in Teaching & Learning. Virtual reality lab, Office of the Provost, University of Illinois Urbana-Champaign. URL: <https://cit1.illinois.edu/cit1-101/instructional-spaces-technologies/armory-innovation-spaces/virtual-reality-lab>. (Last Accessed: May 4, 2021).
- [11] Free3D. Mining rotary drill v3 3D model - Free3D. URL: <https://free3d.com/3d-model/mining-rotary-drill-v3--675841.html>. (Last Accessed: April 8, 2021).
- [12] Exit Games. Photon unity networking 2. URL: <https://doc-api.photonengine.com/en/pun/v2/index.html>. (Last Accessed: January 7, 2020).
- [13] D. Helgason, J. Ante, and N. Francis. *Unity - Video Game Engine*. (Last Accessed: January 7, 2020). Unity Technologies. Unity Technologies, San Francisco, 2019. URL: <http://www.unity3d.com>.
- [14] HTC vive, October 2019. URL: <https://www.vive.com/us/>.
- [15] Philip Kortum. *HCI beyond the GUI: design for haptic, speech, olfactory, and other nontraditional interfaces*. In Elsevier, 2008, pages 107–137.
- [16] NYC Media Lab. 14 virtual and augmented reality projects emerging from nyc media lab this spring, March 2017. URL: <https://medium.com/@nycmedialab/14-virtual-andaugmented-reality-projects-emerging-from-nyc-media-lab-this-spring-af65ccb6bdd8>.
- [17] Eike Langbehn and Frank Steinicke. Redirected walking in virtual reality. In Newton Lee, editor, *Encyclopedia of Computer Graphics and Games*, pages 1–11. Springer International Publishing, Cham, 2018. ISBN: 978-3-319-08234-9. DOI: 10.1007/978-3-319-08234-9_253-1. URL: https://doi.org/10.1007/978-3-319-08234-9_253-1.
- [18] Joeseeph J LaViola, Ernst Kruijff, Ryan P McMahan, Doug A Bowman, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2nd edition, 2017. ISBN: 978-0134034324.
- [19] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Pouperev. *3D user interfaces theory and practice*. Addison-Wesley, 2017.
- [20] J. LaViola, JR., E. Kruijff, R. McMahan, D. Bowman, and I. Poupyrev. *General Principles of Human-Computer Interaction*. Addison Wesley, 2nd edition, 2017. Chapter 4.
- [21] J. Linowes. *Unity virtual reality projects*. In Packt Publishing, Birmingham, UK, 2018. Chapter 7: Locomotion and Comfort, pages 201–235.
- [22] N. E. Miner and S. A. Stansfield. An interactive virtual reality simulation system for robot control and operator training. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1428–1435 vol.2, May 1994. DOI: 10.1109/ROBOT.1994.351289.

- [23] S. Narang, A. Best, and D. Manocha. Simulating movement interactions between avatars agents in virtual worlds using human motion constraints. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 9–16, March 2018. DOI: 10.1109/VR.2018.8446152.
- [24] Alexander Novotny, Rowan Gudmundsson, and Frederick C. Harris, Jr. A unity framework for multi-user VR experiences. In Gordon Lee and Ying Jin, editors, *Proceedings of 35th International Conference on Computers and Their Applications (CATA 2020)*, volume 69 of *EPiC Series in Computing*, pages 13–21. EasyChair, 2020. DOI: 10.29007/r1q2. URL: <https://easychair.org/publications/paper/59bX>.
- [25] PyImageSearch. Pyimagesearch - you can master computer vision, deep learning, and opencv. URL: <https://www.pyimagesearch.com/>. (Last Accessed: April 8, 2021).
- [26] Nico Reski and Aris Alissandrakis. Open data exploration in virtual reality: a comparative study of input technology. *Virtual Reality*, 2019. DOI: 10.1007/s10055-019-00378-w.
- [27] Kate A Spitzley and Andrew R Karduna. Feasibility of using a fully immersive virtual reality system for kinematic data collection. *Journal of biomechanics*, 87:172–176, 2019. ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2019.02.015. URL: <https://doi.org/10.1016/j.jbiomech.2019.02.015>.
- [28] Walker Spurgeon. *Exploring Hands-Free Alternatives for Teleportation in VR*. Master’s thesis, University of Nevada, Reno, August 2018. URL: <https://scholarworks.unr.edu/handle/11714/4545>. Department of Computer Science and Engineering.
- [29] Jeremy Stangroom. One-way ANOVA calculator, plus tukey hsd. URL: <https://www.socscistatistics.com/tests/anova/default2.aspx>. (Last Accessed: October 10, 2019).
- [30] Bearded Man Studios. Forge networking remastered. URL: <https://github.com/BeardedManStudios/ForgeNetworkingRemastered>. (Last Accessed: January 7, 2020).
- [31] Unity Technologies. Unity. URL: <https://unity3d.com/>. (Last Accessed: October 10, 2019).
- [32] Unity Technologies. Unity asset store - the best assets for game making. URL: <https://assetstore.unity.com/>. (Last Accessed: December 30, 2019).
- [33] Us10038894b1 - three-dimensional scene reconstruction from set of two dimensional images for consumption in virtual reality. URL: <https://patents.google.com/patent/US10038894B1/en>.
- [34] Us10230904b2 - three-dimensional, 360-degree virtual reality camera system. URL: <https://patents.google.com/patent/US10230904B2/en>.

- [35] Us10249088b2 - system and method for remote virtual reality control of movable vehicle partitions. URL: <https://patents.google.com/patent/US10249088B2/en>.
- [36] Us20180020204a1 - data structures and delivery methods for expediting virtual reality playback. URL: <https://patents.google.com/patent/US20180020204A1/en>.
- [37] T. Weißker, A. Kunert, B. Fröhlich, and A. Kulik. Spatial updating and simulator sickness during steering and jumping in immersive virtual environments. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 97–104, March 2018. DOI: 10.1109/VR.2018.8446620.
- [38] C. Wienrich, K. Schindler, N. Döllinger, S. Kock, and O. Traupe. Social presence and cooperation in large-scale multi-user virtual reality - the relevance of social interdependence for location-based environments. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 207–214, March 2018. DOI: 10.1109/VR.2018.8446575.
- [39] Patryk Zatylny. Fantasy forest environment - free demo, 2018. URL: <https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361>. (Last Accessed: October 10, 2019).
- [40] Q. Zhou, G. Hagemann, D. Fafard, I. Stavness, and S. Fels. An evaluation of depth and size perception on a spherical fish tank virtual reality display. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2040–2049, May 2019. DOI: 10.1109/TVCG.2019.2898742.