A Drop of Water

- A VR Experience focussed on VR Sickness -

Project Report

19GR1041

Aalborg University
Electronics and IT

Summary

Dette projekt fokuserede på fire forskellige bevægelses metoder der blev analyseret i forbindelse med Virtual Reality (VR) sickness i en VR-simulation af en vanddråbe. Grundfos foreslog dette projekt med baggrunden af, at udvikle en VR Experience (VRE) til uddannelses formål. Dette ønske opstod i forbindelse med de 17 mål "The Sustainable Development Goals (SDG)" som de Forenede Nationer udgav i 2015, til at forbedre menneskelige levevilkår. Grundfos vil dermed undersøge hvordan en VRE, der skal forklare den hydrologiske cyklus vandet gennemgår, kan lære brugeren i den proces der skal til for at rengøre vand.

I forbindelse med bevægelses metoder er der mange forskellige muligheder som vil udvikle mere eller mindre VR sickness for brugeren. Mange forskellige metoder blev udviklet, men kun få er brugt hyppigt i forskellige simulationer. Endnu færre er antaget at være brugbare for scenariet af det frie fald som vanddråbe. De fire metoder denne rapport behandler er: *Teleport*, *Controller*, *Trackpad*, og *Automove*. Teleport metoden udsender en stråle ud fra kontrolleren der kolliderer med den virtuelle verden over en bestemt distance. Efter en knap bliver frigivet bevæger sig brugeren imod den markerede position imens de falder. Controller metoden bruger kontrollerens koordinatsystem i forbindelse med rotation til at bevæge brugeren i kontrollerens retning, indtil de kolliderer med den virtuelle verdens overflade. Trackpad benytter kontrollerens track-pad til at skalere hastigheden brugeren bevæger sig med. Derudover benyttes track-padden til at bevæge brugeren i den virtuelle verden ud fra tommeltottens position på track-padden i kombination med kontrollerens rotation. Metoden Automove bevæger brugeren med en bestemt hastighed direkte imod et mål defineret af simulationen. Udover de forskellige bevægelsesmetoder blev der også lavet 6 forskellige scenarier som vand kunne tage i den cyklus: By, Ørken, Gård, Bjerg, Flod, og Renseanlæg.

I løbet af projektet blev der testet i flere iterationer og hver test havde fokus på forskellige emner. Den første test handlede om indleve sig i den virtuelle verden. Dette blev gjort med forskellige effekter som vanddråber på skærmen i forbindelse med at være over eller under vandoverfladen. Testen have 20 deltagere, hvor i blandt 9 var kvinder. Deltagerne var imellem alderen 20 til 27 og varierede mellem at have ingen til meget erfaring med VR. Gennem testen prøvede halvdelen af delt-

agerne med vand effekterne aktiveret og den anden halvdel uden vand effekter. Efter tiden i den virtuelle verden fik hver deltager et standardiseret spørgeskema "Simulation Sickness Questionere" (SSQ). Det blev fundet at denne effekt forhøjede scoren med omkring 50% af den mængde VR sickness hver deltager følte. Derfor blev effekten fjernet fra de øvrige tests. I forbindelse med test 2-4 blev der testet forskellige muligheder at omsætte de fire bevægelsesmønstre som nævnt før. Hver metode have 2 forskellige versioner baseret på den samme ide og da kun en af hver metode ville blive brugt i den sidste test, måtte den bedste versions findes. De forskellige tests havde hver 22 deltagere for Teleport, 21 for Controller, og 22 for Trackpad. For Teleport blev det fundet at deltageren fortrak at falde med en kurve i steder for et vertikalt fald ned til deres mål. Controller og Trackpad testede om deltagerne fortrak at orienterede sig ud fra kontrollerens rotation eller deres skærms rotation. Det blev fundet i begge tilfælde at kontrollerens rotation var fortrukket, der var dog ikke stor forskel mellem de 2 versioner for Trackpad. I den sidste test blev de bedste versioner af hver bevægelsesmetoder sammenlignet med hinanden til sidtst fik deltagerne muligheden for at prøve en af de 6 scenarier. Det blev fundet at Controller gav deltagerne mindste VR sickness, grundet dens mængde af kontrol og hastighed. Dette virker til at være grundet i dens mængde af kontrol og hastighed. Angående scenarierne der også blev testet kunne der ikke siges meget, da testen endte med at kun blive testet på få deltagere per scenario. Dette betød at hver enkelt deltagers svar havde stor indflydelse på testen og ekstremer ville rykke meget med den indsamlede data. Men der blev fundet at deltagere fortrak at kontrollere deres egen rute over en rute der er bestemt for dem.

AALBORG UNIVERSITY
STUDENT REPORT

Electronics and IT
Aalborg University
http://www.aau.dk

Title:
A Drop of Water
A VR Experience focussed on VR Sickness

Theme:
VR sickness study

Project Period:
Spring Semester 2019

Project Group:
1041 VGIS

Participant(s):
Moritz Alexander Thormann
Niklas Vinther

Supervisor(s):
Martin Kraus

Copies: 1

Page Numbers: 121

Date of Completion:
June 4, 2019

Abstract:

This project was developed in collaboration with the Danish pump manufacturer Grundfos, which proposed a problem of a Virtual Reality Experience, that is supposed to be used for educational purposes. This was done to support the United Nations "Sustainable Development Goals" by visualizing the hydrologic cycle in Virtual Reality. The projects main focus was to investigate four different methods for free fall in Virtual Reality. The methods, which were similar to frequently used interaction methods in other Virtual Realities, were developed and tested. The four methods were named Teleport, Controller, Trackpad, and Automove. Through initial testing it was found that on-screen effects may increase the score of a Simulation Sickness Questionnaire, meaning the users felt more Virtual Reality sickness. The project ended up finding that the more freedom users had the less Virtual Reality sickness they felt and that speed had a high impact on the outcome of an Simulation Sickness Questionnaire. From the four different methods Controller was found to be the most liked by the users. In addition it was found that point to point movement is highly useful and simple but not preferred over freedom.

Contents

Preface

This master thesis project was done in an effort to create a template for how falling could be done without increasing the effects of VR sickness on the users.

Four different locomotion methods have been looked at, though this was far from all the possible options available. Of those four locomotion methods, only two versions of each was created and tested against the other version. The best performing version where then tested against the other three best performing version of their locomotion method.

The report is split up six chapters. Each displaying a different aspects of the process. The first works with the topic why the project was made, Chapter 1. Next is all the background knowledge used for the creation of it, Chapter 2. Chapter 3 holds the design of the project, what tools were used, what was created, and what functions lies behind it all. This was all tested in Chapter 4 where each of the five test is described, calculated, and discussed. The conclusion of these tests can be seen in Chapter 5, which is then evaluated in Chapter 6.

Thank you very much for reading and we hope you find this report interesting.

Aalborg University, June 4, 2019

Niklas Vinther
<nvinth14@student.aau.dk>

Moritz Alexander Thormann
<mthorm14@student.aau.dk>

Glossary

During this report some of words or abbreviations would require deeper explanation, due to that this list will contain the most frequent used abbreviations and terms that would require background knowledge.

- *Collider*, is a Class within the Unity Engine, this class can detect collision and provide cover for the other objects.
- *Framerate*, the frequency of newly generated images displayed on a display, also frequently referred to as Frames Per Second or (FPS).
- *Freelook*, a common used movement method for 3D video games where the orientation of the virtual display is mapped from one input device while the translation is based on a second input device[43].
- *GameObjects*, are the base class for all entities in Unity Scenes [23].
- *Head Mounted Displays*, or in short *HMD's* are devices that provide information on screens that are located in front of the eyes, common known devicese are Oculus Rift[42], HTC Vive [58] and PlayStationVR [45].
- *Immersion*, a deep mental involvement in something [15].
- *Linerenderer*, is a Class within the Unity Engine, this class can render a line, using points and colors or textures.
- *NPC*, or Non-Player Character, a person or creature in a video game that is not controlled by the user [51], [55], [14].
- *Prop*, or props, is a object that acts as a source of assistance for a user.
- *Raycast*, is a Class within the Unity Engine, this class is used to send out a beam from one point along one direction, this can be used to detect intersection with a collider.

- *VRE*, a Virtual Reality Experience. The application or program that can be executed and which works with HMD devices. In this project it refers to the prototype program developed.

- *VR Sickness*, motion, simulator or cyber sickness, this sickness appears for some users and can be caused by mismatch of the body sensors, similar to beeing seasick [3], [44]. This project uses VR sickness as a term for the combination of these three different sicknesses, due to it being a VR simulation. Motion sickness comes from real world motion or the appearance of it. Cyber sickness symptons appearing while using VR. Simulator sickness is for shortcomings in a virtual simulation, such as feeling unwell while driving a simulation but not feeling unwell while driving in reality [29].

Chapter 1

Introduction

This introduction is describing the general origin of this project and how it started. It provides a brief overview of the important aspects that led to forming the solution for the problem.

1.1 Origin

In 2015 United Nations (UN) agreed on 17 goals, called the Sustainable Development Goals (SDG), to help people with peace, plants and poverty [40], [16]. One of those goals, goal 6, is to provide clean water and sanitation on a local level for developing countries within Sub-Saharan Africa, Central Asia, Eastern Asia, Southern Asia, and South-Eastern Asia. Seven sub-goals have been set for SDG 6, one of which is to have accessible clean and affordable drinking water as three in ten people lack access to this basic need [41].

To get a better understanding of what process water goes through to get clean and safe to drink Grundfos suggested making a Virtual Reality experience (VRE) of the hydrologic cycle. In making such a VRE there are different methods of locomotion, methods of moving from one point to another. One way to do locomotion is to instantly move the user to a designated position. Using a controller the user will pick a spot they would like to move to and the simulation will do a quick fade to black and then fade back at the designated position, this is a common way of moving in Virtual Reality (VR) [56], [13]. Instead of fading to black and back again, Oculus VR found that if four planes (two for each eye) would mimic blinking and if they close at the speed of human blinking, the user would filter it out and not notice it [21]. Which would lead to the user feeling like a shift just happens in a blink of an eye.

Another method of locomotion could be for the user to walk around themselves in the physical world. This could then be mapped either directly to the VRE, or the step length could be modified e.g. increasing the virtual range of the physical step.

These two different locomotion methods may not be equally suited for all types of VRE. Creating a VRE with falling as a focus, the teleport locomotion may not be as well suited as using a joystick or detecting if the user is leaning to one side, using the controllers or head mounted display (HMD), to move. Using an ill suited locomotion methods may lead to lose of immersion or an increase of VR sickness.

1.2 Use Case

Grundfos is a Danish manufacturer that produces water solutions such as motors, controllers, and pumps. Focused on clean water, wastewater, and moving water in general, helps Grundfos to connect with millions of people all over the world [1]. Grundfos provided a case in collaboration with the Sustainable development program from the United Nations and Aalborg University. The focus of that case was to generate an educational VRE that should explain the hydrologic cycle. The general challenge for this project was set around the idea that the user in VR was able to navigate through different paths dependent on their choices while representing a water drop, to provide knowledge about how water is treated when it lands.

1.2.1 Application Scenario

The hydrologic cycle, as seen in Figure 1.1, is waters cycle from falling to earth from the atmosphere to evaporating back to the atmosphere. When the water falls to earth it can be absorbed by plants, the soil, land, in different bodies of water, or evaporate while in the air [17]. This is the cycle without any intervention from humans.

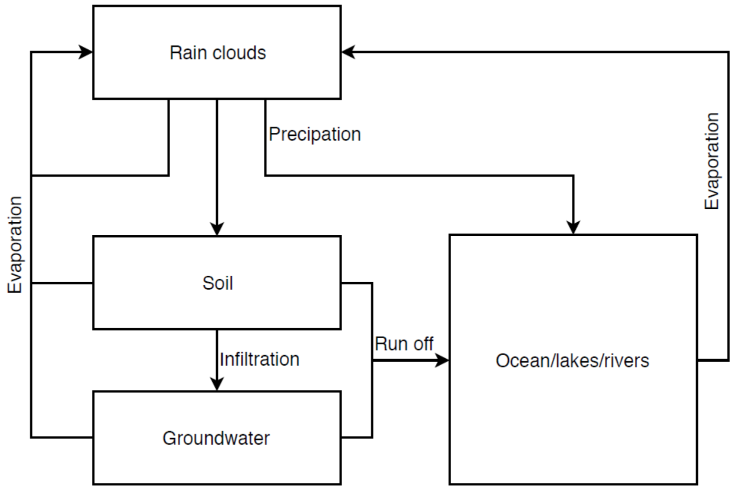

Figure 1.1: The simplified hydrologic cycle. Water falls from the sky, gets absorbed, and evaporate.

If humans, or to be more precise Grundfos, were implemented in the hydrologic cycle, unfiltered water would in some cases be pumped up from the ocean, groundwater, rivers, or lakes. After pumping, the unfiltered water is taken to a treatment plant, here various filters and chemicals are used to clean the water. After the water has been cleaned it is distributed around the area, in water towers, cities, etc. Next step is to collect the wastewater, water that has been used for different purposes and road drain water. Lastly the wastewater is transported to another treatment plant. Here it is again cleaned and the sludge is removed. The clean water is then pumped out into the ocean and the sludge is stored. This process can be seen in Figure 1.2.

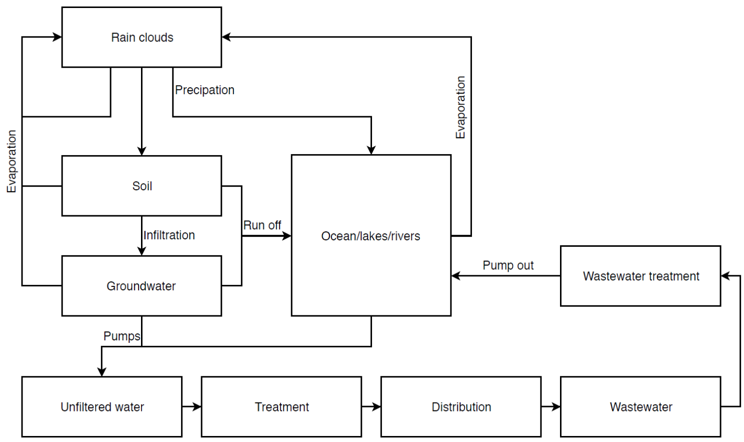

Figure 1.2: The simplified hydrologic cycle, plus Grundfos water treatment and use.

1.3 Problem description

There were multiple components that needed considerations when designing the VRE. For that three points of interest were created, one for the main goals, the second for the requirements to achieve the goal, and the third for the approach taking to achieve the goal.

Goal

- *Find a suitable navigation method for a multi-path Virtual Reality experience based on the hydrologic cycle*

Requirements

- *Minimize VR sickness*. Try to design the VRE in a way that would either remove or minimize VR sickness.
- *Accurately track VR input*. Be able to handle if controllers or HMD loses connection or other wise is not responding proper.
- *Multiple paths*. Allow the user to pick multiple paths based on the hydrologic cycle.

Approach

- *Test multiple locomotion methods*. Find one or more method that are suited for the VRE.

Chapter 2

Background theory

This Chapter consists of the general background knowledge which was gathered in order to build this project. It is designed to reveal what other people have done in this field so far and what they learned that might support a project of this sort.

2.1 Navigation

Navigation is important for a variety of game like applications, since it is mostly necessary to lead the user to their destination from their start point. It consists of the combination of locomotion and wayfinding.

2.1.1 Locomotion / Traveling

As mentioned in Chapter 1, Introduction, there are different locomotion methods for VR. In the article "New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology" [13], 36 articles were reviewed which lead to 11 different locomotion methods in VR. LaViola et al. found some of the same locomotion methods and a few more, in the book "3D User Interfaces Theory and Practice" [18], These methods add up to the following list:

- *Arm swinging*:
  The user swings their arms along side their body while remaining stationary. This movement can either be picked up by a camera or by controllers which will translate the motion into virtual movement. The orientation can depend on where the user is looking or other factors such as front facing direction.

- *Chair-based*:
  Uses the rotating and tilting movements of a stool or chair as an input device. The tilting of the user can be translated to increase or decrease the speed in the virtual movements. Rotating on the stool could be used for rotation in the VRE.

- *Controller/joystick*:
  This was seen in Bernhard et al. [46] and mentioned later in this report. It focuses on having the user control their movements with a controller or joystick, both speed and orientation, such as the "Xbox Wireless Controller" or "Sony PlayStations $DUALSHOCK^{®}4$".

- *Grab-and-pull*:
  Here the user can either grab certain objects or the virtual world itself and then drag themselves in a direction. The orientation can be based on a combination of the direction of the HMD and where they put their hands. The speed could be based on how fast the user pull back their hand(s) towards themselves.

- *Gesture-based*:
  Translating the virtual player by movement from various gestures recorded by a camera or controllers. Such as a push gesture or pointing with one arm.

- *Head-directed*:
  The head orientation is used to control in which direction the user is moving in the virtual world. Speed could be fixed or change depending on the tilt of the head.

- *Human joystick*:
  A sensory board, such as the *Nintendo*® *Wii*™ balance board, is used to translate the users tilting into virtual movement. Tilting to one side will have the user move to that side in the Virtual world and the more the user tilt the faster the movement could be.

- *Lean-based*:
  Following the same principles as Chair-based locomotion method, Lean-based finds the speed and direction depending on how far a user bend and what orientation the user is leaning in.

- *Non-physical-input-devices*:
  This could be devices that detected if the user thought about moving, such as electroencephalography, which measures brain activity. Classification methods for that would generate the input, which will be transformed into movement or orientation, this method would be non-physical based.

- *Steering props*:
  When a VRE takes place inside a vehicle of some sort, the user can utilize physical props mimicking said vehicle's steering mechanism. This could for instance be the cockpit of a plane, the driver seat of a car, or the command bridge of a space ship. These props can help the user control speed and direction of locomotion.

- *Real-walking*:
  Reading the user's movements into virtual movement without alterations of rotation or distance traveled. Meaning walking forward one step in reality equals walking forwards one step in the VR.

- *Redirected walking*:
  The user can walk freely inside a physically limited space available to them, but the VRE uses techniques to introduce an mismatch between the user's real movements and their virtual movements. This can help compress the larger VR world into the limited physical space. The games can feature a maze which would be physically impossible, such as making four right in quick successions would lead to a new place.

- *Representation*:
  Here the user is given a map of the environment. They can then interact with the map, by clicking on a desired position they would like move to, they can then be teleported or their avatar can move towards the desired position. Another way is giving the user a smaller representation of the environment with a small version of their avatar. In order to move the user will pick up the smaller avatar and move it to a new position, they are then themselves moved to said location.

- *Reorientation*:
  Like Redirected walking, the user can move freely around in a limited physical space, while this method modifies the rotation of the user in a variety of ways such as the paper "Towards Virtual Reality Infinite Walking" [4] suggests. Reorientation methods can also be used with the help of blinking or fading as Tom Forsyth at Oculus VR suggested[39] increasing the virtual walking distance. In addition one implementation could also be to let the user walk until he reached the end and then physically turning while virtually staying stable, thus the avatar can continue walking in the same direction.

- *Teleportation*:
  As mentioned earlier in Chapter 1, Introduction, it is using a controller, or other devices, to specify a designated position and then instantly move to said position. It can also be done slower, so the user can see a transition from the start position to the end position.

- *Torso-based*:
  By adding sensors e.g. around the waist, the orientation of the torso can be found. This orientation is then translated into a direction for the VRE, which then has a constant speed.

- *View-based*:
  Like Chair-based, the orientation of the HMD is found and transformed into a direction for the user to move, with a constant or altering speed.

- *Walking-in-place*:
  Virtual movement is achieved by the user walking, jogging, or making a stepping motion in place. The faster the movement the faster the speed in the VRE. The HMD orientation can be used to decide direction of movement.

- *Zero-gain-walk*:
  Sharing a few similarities to both Human joystick and Walking-in-place, this methods allows the user to walk in place without moving forward in the physical space. This can be done with different controllers, for example a treadmill, a low friction surface, or a step base device. A treadmill can be both active, where the user's movement is detected and actively counteracted by the treadmill, or passive, where the treadmill react to the user's momentum and weight. Other solutions would be a surface offering low friction allowing the user to slide on the surface when or after taking a step. Also the step based device, which works by having two plates for the user to walk on could be used. These two plates will predict where the user will put down their foot and catch them.

Some of these methods are similar or do only affect either movement direction or speed. With this in mind, a VRE may needed two or more methods, if they did not provide both translation and orientation for transitioning. An example could be to combine torso-based and lean-based, so that lean-based control the speed and torso-based control the direction of travel. The methods would have required to be combined but could also be used individually in the same VRE, for example when using a combination of real-walk and steering props. This could be the case where a user would be required to both fix and drive a car.

2.1.2 Wayfinding

For a user to perform wayfinding they need different cues and aids, it is key to assist the cognitive process for determining where to go [18]. With these the user can create a mental map or navigate their surroundings. There are two kinds of cues, User-centered and Environment-centered, which supports the user in different ways.

User-centered cues are *field of view* (FOV), *motion cues*, *multisensory output*, *presence*, and *search strategies*, as the names suggest these are cues that are centered around the user. The FOV is the width and height of the users vision, the less FOV a user has, the more they have to move their head to see the world around them, Figure 2.1 displays the FOV. If too large a fisheye effect can appear. A low FOV can lead to VR sickness and a lack of optical-flow, which can be a strong motion cue. Having a lack of motion cues, such as vestibular cues, can also lead to VR sickness. One way to get those cues are by walking-in-place as it can create a connection between visual and vestibular inputs. Other sensory inputs could also be implemented, such as auditory, tactile, taste, and so on, which could all increase the sense of Presence. The sense of Presence is sense of being present inside the VRE, what is also called immersion in this project. Lastly there are different search strategies, which is how users will search in the VRE. A user trained in searching will apply different strategies than an untrained user [18]. Where an untrained user may only use big landmarks, a trained user could also use patterns in the environment such as paths/roads to find their way around.

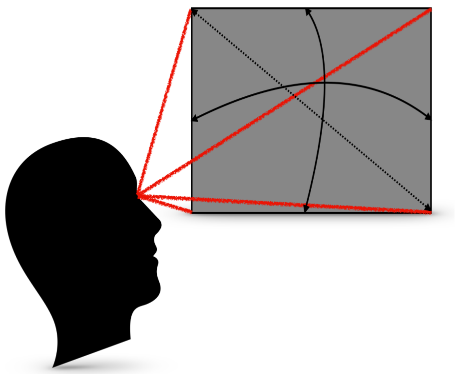

Figure 2.1: The Field of View displayed, the dotted line displays the field, while the solid black lines display width or height of the field.

Environment-centered cues are cues from the environment, such as *environment legibility*, *landmarks*, *maps*, *compasses*, *signs*, *trails*, and *reference objects* a combination of those procedures can be found in Figure 2.2. When designing the environment it should be legible for the users. Meaning it should be possible for the user to get an understanding on how the world is set up, an example could be a city. The user should be able to learn how to find their way around in that city. One way to help this is by implementing landmarks. Though a problem with implementing landmarks is that it is a vague term. LaViola et al. describes landmarks as easily distinguishable objects [18], such as a major building or a sculpture that stands out from its surroundings. But a landmark can also be a minor location, something that only the user would recognize. In the real world this could be a workplace, the user's home, or a bench in a park, anything that is somehow worked into the user's activity patterns [24]. In a VRE it could be a place where the user or a group previously were ambushed, a place they found something, or a corner they often run by to give some examples.

Figure 2.2: The minimap of the game GTA: Vice City [26], this minimap combines a variaty of the named features, such as a compass, a map, the viewing direction and even trails if a position is defined by the player.

A landmark should either be a local- or global-landmark as they can be used for different things in the VRE. A global-landmark, as the sun, should be visible from most of the environment or big parts of it, so that they can help orientate the user. In contrast a local-landmark only needs to be visible from a certain position, to help the user make a navigation decision at that given moment. A common Environmental-centered cue is a map, as it can give information of the surrounding environment. Being in a digital media, the map can update along the way, have a highlight of where the user is, such as a marker on a GPS, give positions on landmarks, and more. Along with a map, the user could be provided with a compass. The compass would tell the direction of the target and the map would tell what is in that direction. A compass can also be used on its own to show the direction of the target. Signs, trails, and paths in the VRE can also be used by the user as wayfinding clues. Lastly having objects the user can relate to in regards to the size of it in the real world can provide depth cues to the user in the virtual world. If the user sees a bench, which is located in the distance, that can make it easier to estimate the distance from the user to the bench.

2.1.3 Navigation related work

Looking at some of the Navigation methods in action, Medeiros et al. [37] tested the effects of three different target-based locomotion methods with regards to speed, travel time, and transitions. The three methods were chosen on the base of reducing fatigue and VR sickness when compared to point-steering traveling methods. The three methods were Teleport, Linear Motion, and Animated Teleport box. The first method would move the user instantly to the decided position, the second method moved the user with a constant speed between 30m/s and 50m/s to the decided position, and the third method would do a 1,5 second animation where the user would be surrounded by a box and then instantly moved to the decided position where the box would disappear. There were 20 users, two female, with an age mean of 24, seven users had previous experience with VR. It was concluded that Teleport had the least amount of discomfort among the users and the Animated Teleport box did not affect performance or VR sickness. Other traveling methods, such as real world walking, can also reduce VR sickness but can increase fatigue.

The Gauntlet is a way to travel around a VRE, suggested in the article "Gauntlet: Travel Technique for Immersive Environments using Non-dominant Hand" [54]. The methods let the user move around a VRE using their non-dominant hand whilst begin able to interact with the environment using their dominant hand. Using a motion detection software, the project could detect if the user had closed their non-dominant hand into a fist, this first detection was then set as default or neutral. The user then displaces the first from the default position to move around the VRE. In an effort to minimize fatigue the method can rotate the user's viewport alongside the rotation of the HMD.

Instead of the using the controllers, or other hand-held device, to travel around the VRE, Lee et al. [34] used a walking in place method. Where other methods have used Inertial Measurement Units to detect if a user was walking in place, this method used only position and rotation of the HMD. For testing the novel method they had nine users, two female, with a mean age of $28,56$. The nine users had to do three tests, test one moving forwards, test two moving backwards, and test three error detecting. The first test had the users move forward in the VRE by doing the walk in place motion. The second tests had the user preform the same motion but with their heads tilted upwards, over $30°$. The third test had the users do squats to see if the method would recognize it as a step. In the end the method had an accuracy score of $99,32\%$ and an error rate of 0% from squats.

Lastly another method of locomotion was done in the article "Do We Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice" by Bernhard et al. [46]. In their VRE they had three different ways for a the user to move about.

- The first being walking so that the real movements would match those in the VRE.
- The second was to have a joystick which would move the user around the map but they would still have to rotate and move up and down by themselves.
- The third way was to have the joystick control all movement, including rotation and up-down movement.

From testing it was found that the first locomotion method, real world walking, had better performance benefits compared to the two other methods. The first method also showed an increase in viewpoint changes and shorter navigation paths compared to the others. This could suggest *"a rotation/translation tradeoff and different navigation strategies"* said Bernhard et al. The second method, joystick translation and physical rotation, was better at performance compared with the third method, almost to the point of comparable performance in searching efficiency and time with the first method. They ended up concluding that there can be considerable navigation improvements from allowing full-body rotations, which would lessen cost, space, tracking, and safety requirements normally associated with real world walking.

2.2 VR sickness

One of the ways to demonstrate VR was with roller coaster rides, where a user would explore a 360° experience that can be obtained with HMD's orientation. For some time videos were people lost control over their body tension have been available on the internet. This loss of control however is strongly related to VR and not necessarily to the user [29].

VR sickness with its symptoms of discomfort, nausea, sweating, disorientation and more [31], can be caused by a variety of factors, most of which correspond to conflicts between the humans biological sensors and the virtual perception, such as motion which frequently results in a miss match between real and virtual motion [29].

In addition, VR sickness can also be caused by the simulator itself in this case the HMD, or the computer generating the images for the HMD. The most commonly known issue is delay and frame rate related [61], [28], [18], [39]. The frame rate issue, that is caused when the computer is not able to render enough images per second, can commonly lead to headaches and generally lead to VR sickness related symptoms. The delay latency however can lead to an miss match between the physical input and the virtual conversion, thus things can appear delayed which can trigger the VR symptoms [5], [29].

In order to avoid VR sickness a variety of solutions have been developed. MONKEYmedia®, a research and development company committed to user friendliness in technology [12], developed such a method. MONKEYmedia® produced a product called BodyNav, which focused on natural movement in order to avoid VR sickness, by utilizing changes in rotation of the head and torso to generate movement in VR [47], [38]. In order to ensure that the user still has the possibility for changing the viewing angle freely in the virtual environment only a limited set of degrees, which can be tweaked, are used for the movement method [12], [38].

Bangay and Preston [9] tested two VRE, "Swimming with dolphins" and "Roller coaster", to see if there was an effect on the heart rate and VR sickness. Focusing on the roller coaster experience, it was tested on approximately 750 people with

143 completing a questionnaire afterwards. The spread of people was 63 male and 80 female, between the age 5-54 with an mean of 18 years. Some of the questions in the questionnaire asked about tendencies to VR sickness, expectations, control in the VRE, VR sickness experienced, and excitement. It was found that excitement, comfort, sound, image quality, and age had an effect to users immersion. Whereas immersion had an effect on VR sickness, Control, Excitement and Replayability.

2.3 Virtual Body ownership

To create ownership of a virtual humanoid avatar, Maselli and Slater found, in the article "The building blocks of the full body ownership illusion" [36], that the avatar needs to have a realistic skin tone and take up approximately the same amount of space in the virtual world as in the real world from a first person perspective in order for the user to feel it is their body. This was done with three different tests, test one to see the effect of visuotactile and visual sensorimotor stimulation. Test two to see if visual perspective was a critical factor for full body ownership illusion. And test three to see the impact of a plastic mannequin avatar compared to a human avatar, with regards to realism of appearance. For the three tests they had 54 users, with 36 users for test one. From all the tests it was found:

- A first person perspective was necessary.
- Seeing a realistic body in the correct place and position can be enough to provide body ownership.
- The more realistic skin and cloth are, the stronger body ownership is.
- Multisensory and/or sensorimotor contingencies can have a strengthening or weakening effect on body ownership.
- Body ownership can effect the way touch is perceived. This effect is increased with a first person perspective and realistic skin and also clothing.

Looking at a non-humanoid virtual avatar, Lee at al. [33] found that the way the avatar interacts with the world around them had a bigger impact on creating body ownership than the appearance of the avatar. In the article "Being Them: Presence of Using Non-human Avatars in Immersive Virtual Environment", users would move an apple around in a virtual environment using either a human, wolf, or snake avatar. The human could use both hands to interact with the apple, the wolf could push the apple around using its two front paws, and the snake would pick up the apple with its tail. They had 22 users, one female, and did one-way ANOVA tests with a 0,05 significance level. In their results they found that there was no significant difference in acceptance and control the three different avatars, but there was a significant difference in appearance from their normal body. Of the three avatars it was found that the wolf showed the highest significant difference. Due to the wolf's different interaction method it is believed that the users felt they became more like the avatar over time. As there was no significant difference between the human and snake avatar, with regards to change of interaction, it was believed that the interaction method has a greater impact on virtual body ownership than appearance.

2.4 Replayability

There are different ways to obtain or increase replayability in a video game, in the book "Game Design workshop, A Playcentric Approach to Creating Innovative Games" [22], by Tracy Fullerton, multiple methods are suggested. One way is to have multiple solutions to a problem, another is to make non-player characters (NPC) feel alive by providing them with behaviours, and thirdly is to give multiple routes to the final goal.

Rafael Ballagas and Steffen P. Walz [8] suggested in 2007 a novel idea for tourist to explore the history of Regensburg, Germany. In an effort to get the tourist to keep playing and coming back to the game several methods were implemented. The first method was to have more characters to interact with than the user can reach in the limited amount of game time. Secondly, when the game ends the user's statistics was saved so the next time they play they could pick up where they left off. Thirdly, the game provided non-linear, self-directed routes to the target which could give the user a new unique route each play through. Lastly, each character the user could interact with had multiple quests for them to complete, leading to a reason for the user to revisit the same character multiple times.

The paper "Increasing Replayability with Deliberative and Reactive Planning" [35] suggested an adaptive behaviour for enemies in video games could also increase replayability. In the paper it is stated that without tactical adaptation for an AI enemy the user will simply learn one strategy and win every conflict by simply applying that strategy over and over again without thinking. Giving the AI enemy the ability to adapt would challenge the user to use their cognitive skills.

Replayability can also be improved by *Grip and Grind*, terms that Chris Bateman introduces in the book "Imaginary Games" [11] to describe the effects that can make a user repeat an action in the knowledge or hope of a reward. For Grind the user knows what the reward is and that it will come, whereas Grip is the hope of reward or the feeling of being close to winning. These two terms can be used individually as they are different but Grind can also generate Grip. This happens when a user repeats the same action and knows that a reward will come but not when it will be.

Chapter 3

Materials

This Chapter describes the general materials that were created or used for the given problem that Grundfos suggested and which is described in Chapter 1, Introduction. The decisions made in this Chapter were meant to support the requests of Grundfos and simultaneously develop a prototype product that is able to explore the field of technology, and is as close to a real solution as we could develop.

3.0.1 HTC Vive

The HTC Vive is a highly used HMD and according to Steam Statistics, as shown in Figure 3.1, it is only beaten by the Occulus rift. However the sensors used within the Vive and the general used capability of the hardware in the HMD set the decision for this project to work with the Vive [48].

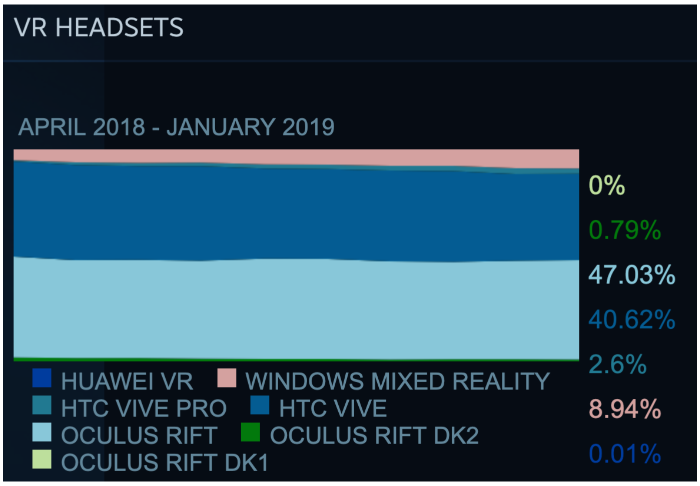

Figure 3.1: The Steam Hardware & Software Survey: January 2019, which provides a overview of how many users work with what hardware. [48]

The HTC Vive uses a 90*Hz* display with a resolution of 2160x1200 pixels and include SteamVR Tracking, G-sensor, gyroscope, proximity as sensors, which utilizes the front camera and the base stations in order to track the user's head [59] and controllers. The way the HTC Vive setup works is by using a set of base stations positioned out in a test area, which then help the internal measurement units to predict the position, orientation and movement of the user [10], by using sensor fusion.

3.0.2 Unity Engine

As a VRE had to be programmed an engine had to be chosen. The "Unity" engine is a game engine developed by *Unity® Technologies* in 2005 as a cross platform real time simulation environment [57]. The engine is supported by a variety of platforms and claims to be the "Industry-leading multiplatform" [20]. It supports *C#* as the main programming language however JavaScript and Boo were supported on some versions and plugins for other programming languages such as $f\#$ have been developed as well. The engine supports not only 2D development, but it is also capable of handling 3D environments. *Microsoft® HoloLens* is a HMD for augmented reality currently only released to companies, for which *Unity® Technologies* states to have produced 91% of applications [32]. In general the Unity Engine is a widely known and well documented program that enables the user to create virtual solutions of all sorts. In addition the Unity engine also contains a store, in which users can contribute with own creations, this can be for free or for a price, defined by the developer sharing his creations.

3.0.3 Steam VR

As this project was able to work on the HTC Vive [58], the project was able to utilize VALVE's SteamVR [49], and its corresponding Unity plugin: "SteamVR Plugin", which was released in April 2015 [50] and is highly useful when developing VR applications in Unity. Steam VR is a software development kit, that establishes a variety of functionalists for general cross reality (XR) not necessarily for the HTC Vive, but for a large set of HMDs and controllers. It is useful to establish a connection between the Unity Engine and the HTC Vive. Through the Software Developer Kit (SDK) it is possible to track different physical interactions of the user within a space used in Unity. Other SDK's for Unity exists as well such as "Virtual Reality Tool Kit" [60], however during this project SteamVR got used, since this is used as a base construct for most of the toolkits.

3.0.4 Autodesk Maya

For generating visual content in the world either open source material was used or 3D objects had to be created. Maya is a product made by *Autodesk*, which is a company specializing *"making software tools for people who make things"* [6]. The program allows the user to 3D-model, render, animate, and add special effects to mention a few of the functionalities. It is a very useful tool for any 3D modeling or creative needs a user could have, which is one of the reasons it is one of the most popular tools in movie industry for modeling, animating, rigging, and more. It has been used in a variety of famous movies such as "Guardians of the Galaxy", "The Hobbit: The Battle of the Five Armies", "Blade Runner 2049". Maya has also been used in the game industry for games such as "Call of Duty" and "Halo" [52], [19], [7].

3.0.5 Functionalities

The world which consisted of mountains, a dessert, a farm area, a city, and the water treatment plant had to be made explorable and needed a variety of functionalities in order to feel more alive. Moreover the user was able to move around with different locomotion methods which were described in Section 3.1, Movement Methods. General functions and hereby scripts were used for all the input methods:

- TriggerScene.cs
  This script would trigger a scene switch within unity. Which used SteamVR's *SteamVR_LoadLevel.Begin()* method for changing the scene depending on which *collider* the user interacted with. This meant that if the user collided with the city's ground they would trigger the city scene, which was a new 3D environment, while an interaction with the water treatment plant would trigger the water treatment scene. In case of the city, the new scene would be used for displaying how a water drop was absorbed in a city environment and what happened to it. In addition to avoid clipping through terrain or hitting with the virtual camera the floor, the camera faded to black when the TriggerScene script was activated. When the FinalTest.cs was enabled this script triggered a scene defined by the FinalTest script.

- Indicator.cs
  The Indicator script was used to rotate a 2D texture with identification hints towards the user's head. More so the script was also used to swap the text depending on the identification of the target position. This can be seen in Figure 3.2.

- RestartForTesting.cs
  This script was used to reset the application to the start setup, when the application behaved unexpectedly. Such as when testing there were some unidentified issues initially, that caused the user not to see their controllers, or the interaction of the controllers was not triggered. For those circumstances the test facilitator had to be able to restart the application by reloading the scene and discarding participants data.

- SlowerWhenClose.cs
  Participants that were flying close by objects often underestimated the speed they were going with since over long distances it is not recognized as being as fast compared to close by. When objects were close by, it would be similar to sitting in a train or plane and focusing on object in the distance, which were not necessarily close by. The code for this can be found in the appendix I.

- sphereIndicatorbinder.cs
  This script enabled the different identification spheres above one of the controllers as seen in Figure 3.11 which shows the outside of it and Figure 3.12, this was done to provide a feedforward similar to "The Lab" [53] or "Earth VR" [25], which provide an idea of how the scene around the participant will change if they interact with that particular area or object. This script and the gameobjects under influence were only enabled when using the teleporting methods.

Figure 3.2: The Indicator that recognized the city as the current go to point.

3.1 Movement Methods

For the prototype multiple methods for handling user input was necessary to decide how a user should fall. During development of the prototype the focus was on four locomotion methods, this Section will describe the methods, and how they work. The interaction was triggered by different ways dependent on the locomotion method. In addition to that, as roller coaster scenarios in VR showed, drastic movement could lead to loss of body tension. This was discussed in Section 2.2, VR sickness, which could lead the users to hurting themselves. In order to try to avoid those scenarios when falling as the raindrop from the cloud to the ground, the interactions would be limited for some of the locomotion methods. This project looked into how to perform movements which appear commonly used in video games which might reduce the impact of VR sickness. In order to lessen or all together avoid VR sickness, as described in Chapter 2, Background theory, there were a lot of requirements for the setup. Not only movement wise but also performance related. In regards to the performance it was necessary to keep a stable frame rate, since changing frame rates or drops in the rate could lead to VR sickness due to a miss match between physical and virtual movement, as described in Section 2.2, VR sickness. Thus one of the requirements were for the prototype keep a stable frame rate of at least 45 fps.

3.1.1 Teleport To DropOff

The Teleport method could have been done in many different ways, two of those ways are suggested here, Linear movement and Bézier movement. This locomotion method would use the user's interaction with a controller and thereby a button press would trigger a *raycast* in the direction of the controller, which, when the user released the button, would move the user in one of two ways as shown in Figure 3.4 and 3.5. The *linerenderer* component which Unity provides, was used as a feed forward indication to where a user would move to. The linerenderer had initially a unintended behavior, this was due to the fact that each segment of the linerenderer was set up in such a way that the displayed texture had wrong alignment resulting in an issue as visualized in Figure 3.3. When this alignment issue arose with a rotated head it seemed to clip through the physical position of the user, which was discovered doing testing when some of the participants voiced that they were uncomfortable with this. Also some of the participants suggested that the steps on the Bézier curve where too drastic, so the amount of move to points and displayed points were increased from 10 to 100 for the finial version, smoothing the path.

Figure 3.3: The displayed path issue, the offset was created due to wrong alignment.

- **Linear Movement**:
  At a position where the raycast intersects with the world's colliders the VRE would display a mesh as the indicator for the drop position. When the button was activated it would draw a path, as shown in Figure 3.6, to the drop position, then after releasing the button, the user would be moved above the drop position, along the path markers. When the position above the marker was reached the user would be dropped and fall down to the marked position, this can be seen as the orange line in Figure 3.4.

- **Bézier Movement**:
  The second method for telepot had the user constantly falling down in a curve towards the marked position, instead of moving above it and then falling down. For this the PathViewer was altered, drawing a Bézier curve to the goal position. The curve can be seen as the orange line in Figure 3.5

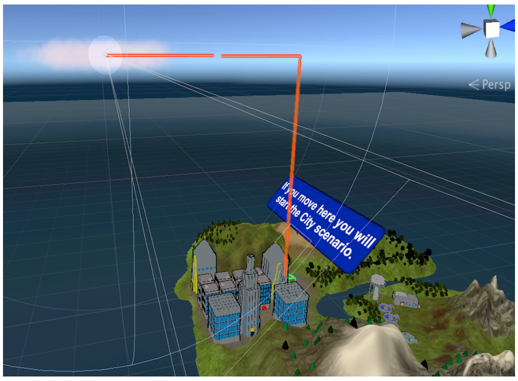

Figure 3.4: Linear movement. The path that the user would take, supported by a linerender indicating the movement form the user's position to the marker position.

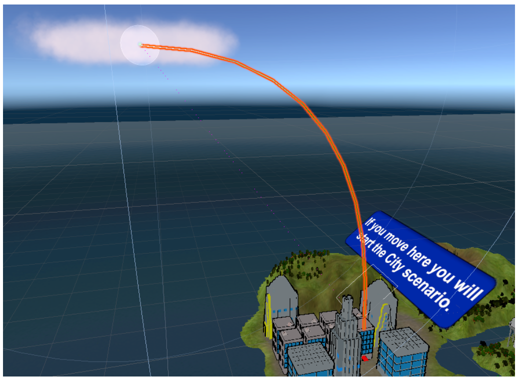

Figure 3.5: Bézier movement. The movement in this method interpolates between multiple points resulting in a constant drop towards the goal.

Figure 3.6: The view on the world from the start position, included the path indication when the teleport button was pressed.

3.1.2 Controller movement

This locomotion method could be based on various factors and move the user differently for each factor. For this project two different ways were looked at, either movement based on the orientation of the HMD or based on the controllers orientation.

- **Display orientation dependent**: This method focused on the orientation of the HMD, for this SteamVR provides the HMD as GameObjects in Unity. The GameObjects in Unity have a Transform component attached to it, which provides the position and orientation. Here the orientation of the controller around their local x-axis were used. When in perfect horizontal position the Euler angle around x is between 359 and 1 degrees as visualized in Figure 3.7.
  For moving forward the controller had to be rotated around the x-axis in an angle of 1°- 89°, while moving backwards was set to be 271°- 359°.
  If this was done, the user was translated along its local z-axis in its orientation direction.

- **Controller orientation dependent**: Initially only the HMD based movement was developed, but after initial testing, flaws were recognized and a second version for this method was created. That version was instead of using the HMD's orientation to translate along, used the orientation of the controller. This resulted in the user moving in the direction that they pointed the controller in. This gave the user the ability to look around the virtual world while continuing to move towards their target.

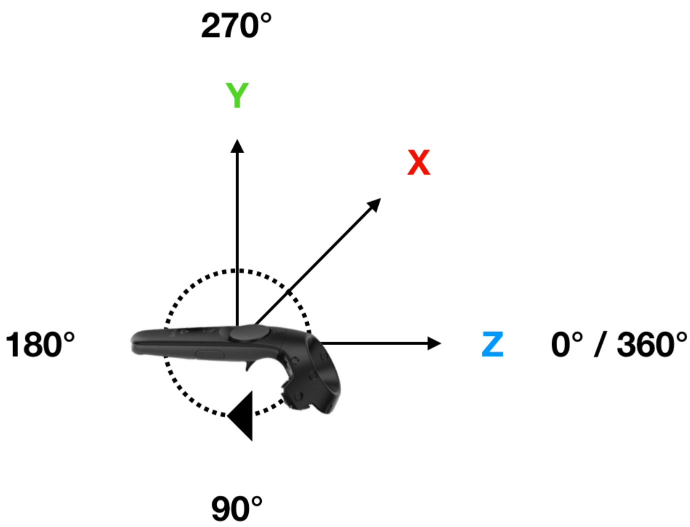

Figure 3.7: The orientation of the controller in a 3D space, for the interaction this project focused only on 1 axis rotation.

3.1.3 Trackpad movement

Like the Controller movement, this locomotion method could be based on multiple factors. Of these factors, for which the position of the thumb on the trackpad and the user's controller / HMD rotation was looked at.

- **Display orientation dependent**: For this movement method, a new "Steam Action", such as a touch recognition, was used, which provided a position of a finger on the touch pad, this position was then used for moving the user in the VRE. The 2-Dimensional vector (X_1, Y_1) of the trackpad was used to define the 3-Dimensional vector for the translation of the user inside the VRE. This was done by using $(X_1, Y_1) \rightarrow (X_2, Y_2, Z_2) = (X_1, 0, Y_1)$, a visualized positioning of coordinate system used for the trackpad can be seen in Figure 3.8. This resulted in the user moving based on their orientation of the display, when the thumb was in the $(0X, 1Y)$ position the user would be translated in a forwards direction. When the thumb was in the $(1X, 1Y)$ position the user was translating forwards-right of the viewing direction. However this mapping of the 2D coordinate to the 3D vector, resulted in the loss of the vectors second coordinate, meaning that the user was only able to move up and or down if they looked in that direction.

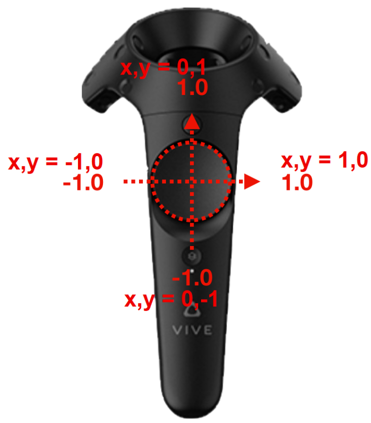

Figure 3.8: The HTC Vive controller with the unit circle identifying the position of the the thumb on the trackpad for input.

- **Controller orientation dependent**: For the second version of the Trackpad movement, the orientation was based on the orientation of the controller. Meaning if the user looked over their right shoulder and held their controller straight out in front of them with their thumb at $(0X, 1Y)$, their would move straight ahead of the torso and not along the display in the direction over their right shoulder. This second version allowed the user to look around the virtual world without the viewing orientation effecting their movement.

3.1.4 Automove

This method was made as a baseline, the user would be moved along a predefined path towards roughly the middle of the generated world. The speed was still depending on distance to objects and the user was still able to move according to the tracked area. The movement in this method was linear.

3.2 VRE Environment

To test the different locomotion methods a simple environment could have been used, such as an empty plane. An environment like that may be easier to control and could have little to no distractions for the user, but it may not provide a whole picture of the locomotion methods when used in a VR game or experience. In an effort to get a more true to life result the decision of creating a more game-like environment, in which to test the locomotion methods, was made to create the prototype. This environment would feature multiple objects, such as trees and buildings, in order to create a packed virtual world which could distract the user. This distraction may cause VR sickness, but it would allow all locomotion methods to be tested in a more realistic scenario, which was deemed to outweigh the negatives. Filling the VRE with objects also gave the user a chance to create landmarks, in the form of different areas such as a city, farm, or desert, which would be used to indicate a different scenario to explore, as described in Section 2.1.2, Wayfinding. The landmarks were seen as global landmarks as they were visually large and can be seen from most of the main world. The game-like world created for this project had multiple other scenes to create a multipath VRE. Those scenes were based on different situations water can experience in the hydrologic cycle. The starting position of the user was decided to be in a cloud, looking down on the world, which can be seen in Figure 3.9.

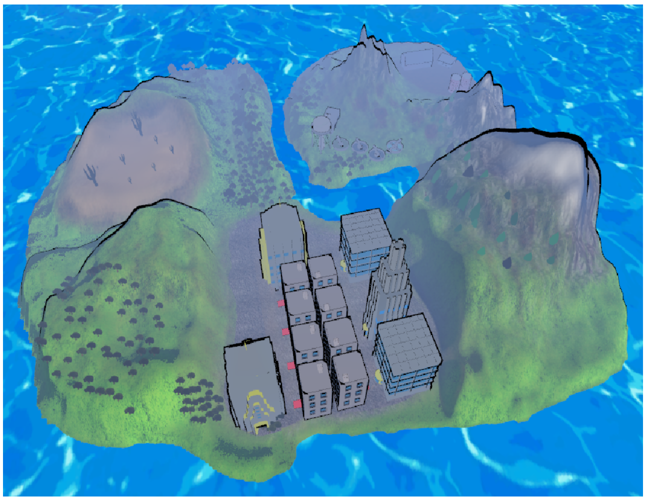

Figure 3.9: A first view on the world from the starting the VRE.

3.2.1 World

This project wanted a cartoonish style to avoid the realism perspective when using textures or colorizing surfaces. The created world thereby looked as visualized in Figure 3.10, this style made the low polygon objects fit into the world. The low polygon objects were preferred as it would lessen the strain on the computational power and help keeping a higher stable frame rate.

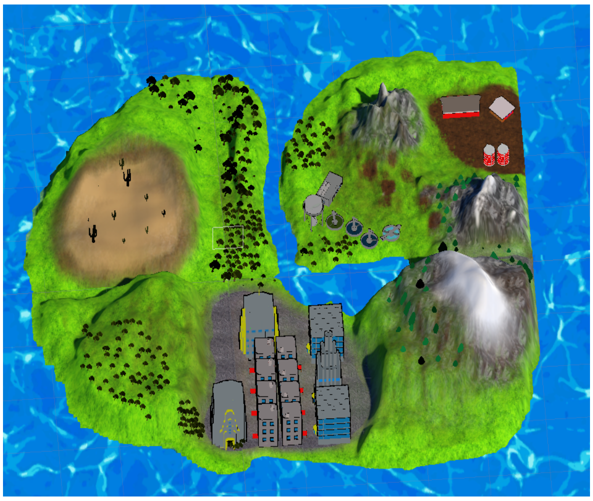

Figure 3.10: The world as seen from above with the adapted water and a in general toonish style.

The water was initially based on the standard WaterBasicDaytime prefab as seen in Figure 3.4 or Figure 3.5, from Unity's internal Environment plugin during the initial locomotion tests. This was changed to a simpler method, using a shader that used a texture as the waters in Figure 3.10 and a noise grayscale texture, that was used to manipulate the displayed texture, dragging in parts of the texture with different strengths depending on the grayscale value of the noise texture. Moreover the water received an additional script changing the height of the water constantly between 0.9 and 1.4 within a 10 second period adding.

Other VR applications such as "The Lab"[53] or "Google Earth VR" [25] implements a globe as a indicator for where the interaction with that globe or a pointer would lead. In addition this sphere could be moved over the head to enable a secondary view of the future location, this is showcased in Figure 3.12. In case of Google Earth VR the interaction from the top view on the earth to the streetview is indicated by a globe on top of the remote, also that globe is visualizing what can be seen from the new perspective. In case of The Lab, the indicator is also visualizing where the the interaction will lead to, however this sphere is also interactive to such an extend that the user has to hold their head inside the sphere to trigger the transition, visualizing more and more of the next locations perspective. In the second version of this project a feed forward identifier, in the form of an orb, had been implemented to provide the user with a better estimate of what to expect when they interacted with certain regions. This indicator can be seen in the Figure 3.11.

Figure 3.11: The indicator sphere that is used as an feedback method to display what scene the user would interact with if they moved there, in this case the Dessert scene.

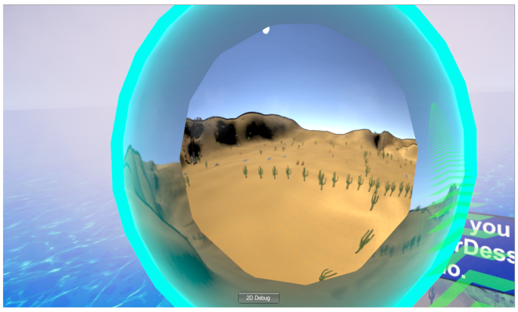

Figure 3.12: The second functionality of the indicator sphere, which provides a 360° view of the future scene, if the HMD position is located in the position of the sphere.

3.2.2 City

The city scene created for this project holds just one of the many different routes water can interact with a city in the hydrologic cycle. It is displayed in the Figures 3.13 and Figure 3.14 These other routes could be drinking water, toilet water, or shower water to mention a few. The route chosen for this project was hitting the street and getting washed down the sewer. The scene holds two levels, first the user hitting the street sliding towards a sewer grate and second they fall down into the sewer where they travel along the water. The route taken was a predetermined route, leaving the user with no influencing of where to go and only standard rotational and translational movement inside the tracked area was implemented to not miss body tension.

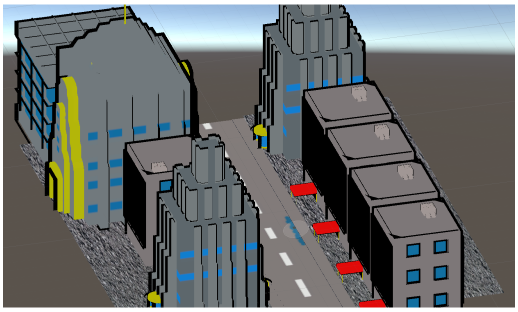

Figure 3.13: An look over the entire City scene.

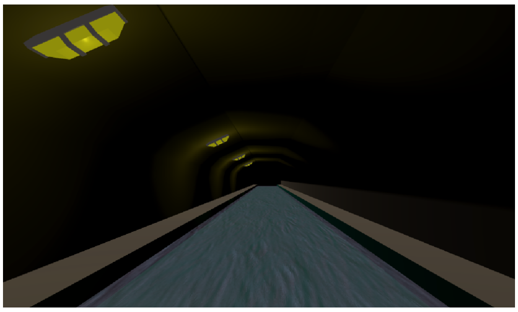

Figure 3.14: View of the sewer part of the City scene.

3.2.3 Dessert

In the Dessert scene the user were dropping from the sky onto the sand. The scene had a lot of details to distract from the particles constantly emitting from the close area around the player. After a while the person would start lift off and vaporize into the sky. Throughout the time spend in the scene the user will experience a vignette effect getting weighted more and more, also chromatic abbreviation will increase during the time. The comparison can be found in Figure 3.15 and Figure 3.16. The distance will have a heat distortion filter on top of the rendered distance.

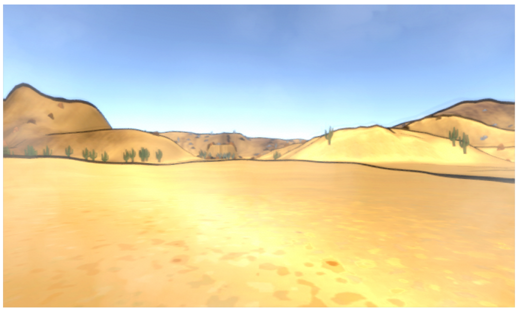

Figure 3.15: A screenshot of the initial Dessert scene.

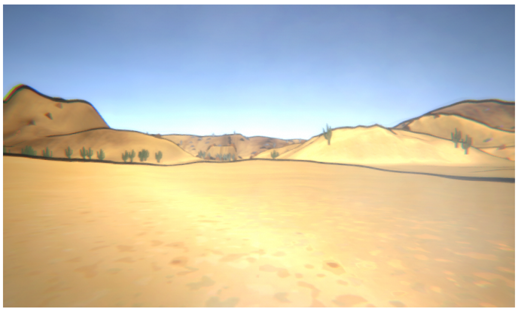

Figure 3.16: View on the dessert later in time, with increased chromatic abbreviation and other effects.

3.2.4 Farm

Similar to the City scene, the Farm scene only had one out of the multiple routes water could do. The route took the user down to the ground fast, as they were falling from the sky, there they went through the ground marginally slower and made their way up a nearby tree, which can be seen in the middle of Figure 3.17. Reaching the top of the tree the user continued up as they evaporate and their screen faded to black.

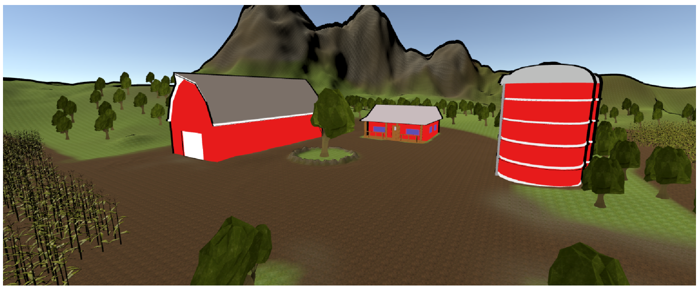

Figure 3.17: A screenshot of the Farm scene

3.2.5 Mountain

In this scene a user was dropped into a river that continues from the top of the mountain to the bottom, the user was pushed around by the shape of the mountains riverbed, and sometimes jumped over the water and sometimes under the water. An overview of the scene can be seen in Figure 3.18 with the river running all the way down. Two different post processing schemes were implemented to manipulate the effect for being under or above water. Thus every time the user would surpass the water level the post processing method would change and thereby providing a more realistic estimation of how eyes would behave when surpassing the water level. This effect can be seen in Figure 3.19, where to the left screen of the user is above water level and to the right where the screen is below water level.

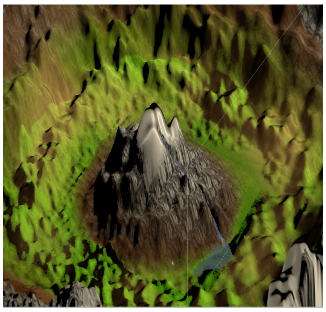

Figure 3.18: The mountain scene that the user is transitioned to if he collides with the mountain in the world-view.

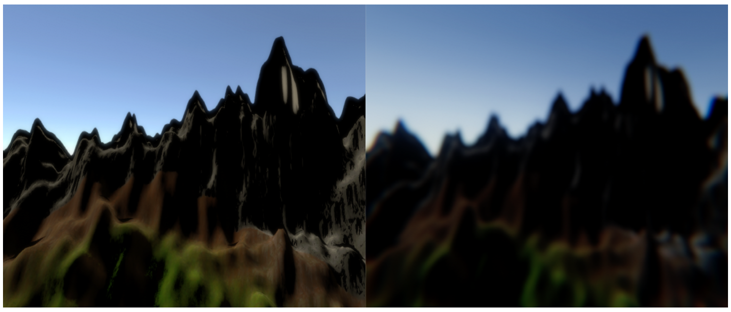

Figure 3.19: Screenshots of the two maximum values, left for when the user is above the water surface, right for when below water surface. When the user surpasses the surface from below it is slowly adapting, while it will instantly change when dropping below the water surface.

3.2.6 River

This scene moved the user in VR along a predefined path, which had positions above and below the water surface, a preview can be seen in Figure 3.20. The user would be moved downstream and was able to move physically in the tracked area thus being able to move the head below the water surface or above, resulting in a change of the post processing similar to the mountain scene. In addition the fog density as well as the fog color changed together with the post processing.

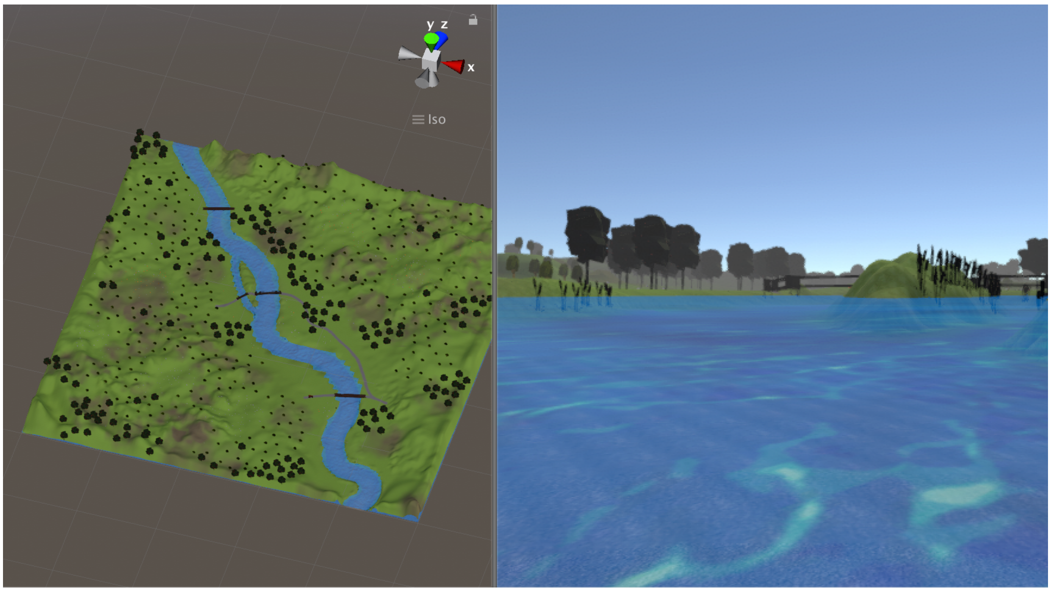

Figure 3.20: A screenshot of the River scene used in the project, with both the scene and the game view.

3.2.7 Wastewater treatment plant

The wastewater scene was a simplified version from the optimal solution suggested by Grundfos [27]. Though major parts were implemented, such as screening the water, primary clarification, Disinfection, and more, all pumping and pressure stations however were left out. This was done in order of simplification and avoiding accelerating. The Figure 3.21 shows a simplified version of one option for purifying wastewater, the solution suggested by Grudfos [27].

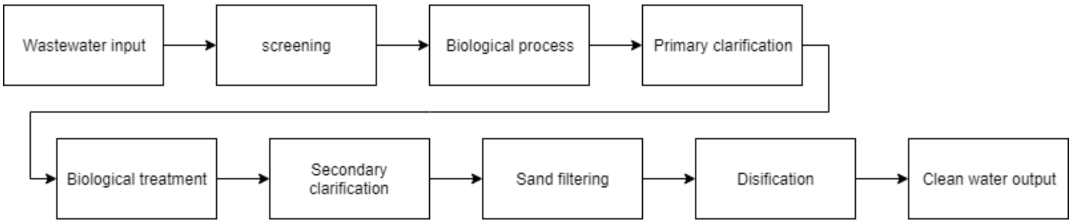

Figure 3.21: A simplified flow chart of the wastewater clarification process.

When entering the scene the user was taken through a pipe to the screening tank and from there the predetermined route through all the other stages. An outside look of the scene can be seen in Figure 3.22. The scene did not take into account the accuracy of placement of each processing stage only the order of the different stages.

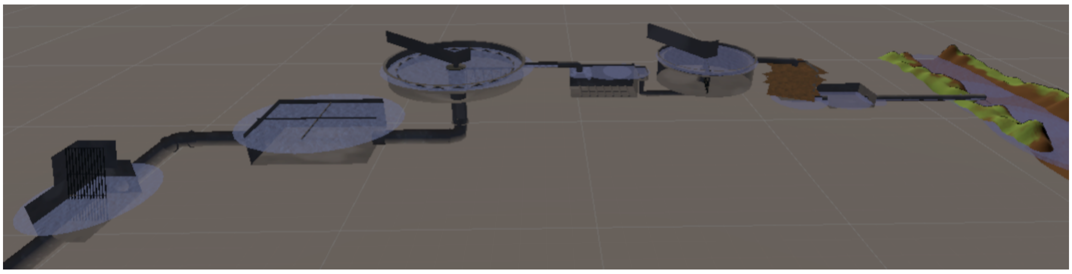

Figure 3.22: A screenshot of the Wastewater scene used in the project.

Chapter 4

Experiments & Results

This Chapter focuses on the tests that were performed during this project. Each test has results, displayed and a discussion. The tests have different focuses dependent on the stage of development, such as:

- Test 1: Water on screen effect.
- Test 2: Teleport.
- Test 3: Controller.
- Test 4: Trackpad.
- Test 5: Final Test.

4.1 Test 1: Water on screen effect

4.1.1 Test 1: Experiment

In an effort to minimize VR sickness caused by sources other than the locomotion methods, a test to see if a Head-Up Display (HUD), consisting of water effects appearing on the screen would have an effect on VR sickness. For this an effect where water drops would appear on screen, and slowly move downwards, while fading out was created. The screen of the participant, when their head came up from under the water, was then displaying the HUD. This effect can be seen in Figure 4.1 where it is enabled and in Figure 4.2 where it is disabled. This effect was made by using a capsule, which was always rotated to be upright in worldspace and which was located at the user's head position. In regards to the size the radius was set to be 20cm, in order to have the HUD slightly away from the display so that the eyes could focus on the shape of the water drops. Also the capsule had a transparent material, this material used a texture that had water drops on it, but the texture was mainly transparent.

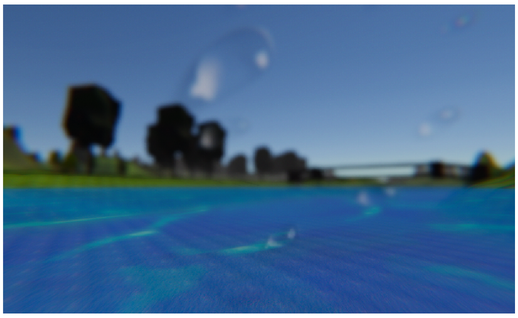

Figure 4.1: Water drops effect enabled. The drops would slowly disappear and roll down the screen when a participant would emerge from the water.

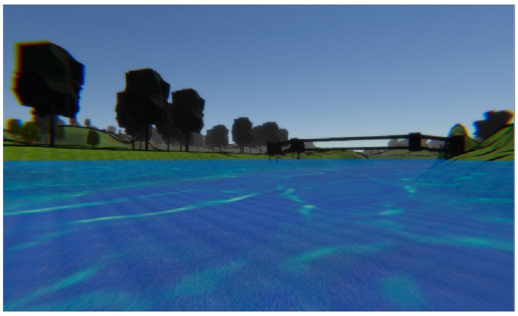

Figure 4.2: Water drops effect disabled. When emerging from the water the screen would be a little blurry, as with the enabled version, but no water drops would appear.

The test had 20 participants, of which nine were female. The participants age spanned from 20 to 27. When the participants were asked if they had previous VR experience it was noted that five participants were trying it for the first time, ten had tried VR a little before, three said they had medium amount of previous experience, and two had tried a lot of VR. For the test a single participant was invited at the time and asked to sit in an office chair, leaving the participant the ability to rotate but not move around from where the chair was placed. When in the chair the facilitator would explain how the HMD worked and how to adjust it to fit their head. When the HMD was placed comfortably the VRE would be started in the River scene, which can be seen in Figure 3.20, where the participants would follow a predetermined route with constant speed and no way to alternate from the route, except by the position they could reach from the chair. The route would alternate between having the user above and below water, triggering the water on screen effect if enabled. For half the participants the effect was enabled while for the other half the effect was disabled, leading to a between-group test. The whole route would take four minutes and 45 seconds to complete and afterwards the participants were asked to fill out a questionnaire about VR sickness, Simulation Sickness Questionnaire (SSQ), which was based on Robert S. Kennedy et al. [30] and can be seen in appendix A

4.1.2 Test 1: Results

The results of the SSQ in this test display that for four of the questions the Enabled version did not score highest. Those questions were 6. *Salivation increasing* and 13. *Dizziness with eyes closed*, with a tie, 10. *"Fullness of head"* where the Disabled version had a higher score, and 16. *Burping* where both versions had a score of 0. For an overview of all scores see Figure 4.3, which visualizes the score for both versions for each question. The biggest differences in scores came in question 4. *Eye strain* with a difference of six and question 9. *Difficulty concentrating* which had a difference of four, where the Enabled version achieved the highest score in both. In question 8. *Nausea* and 15. *Stomach awareness* it was only the Enabled version which got a score, higher than zero.

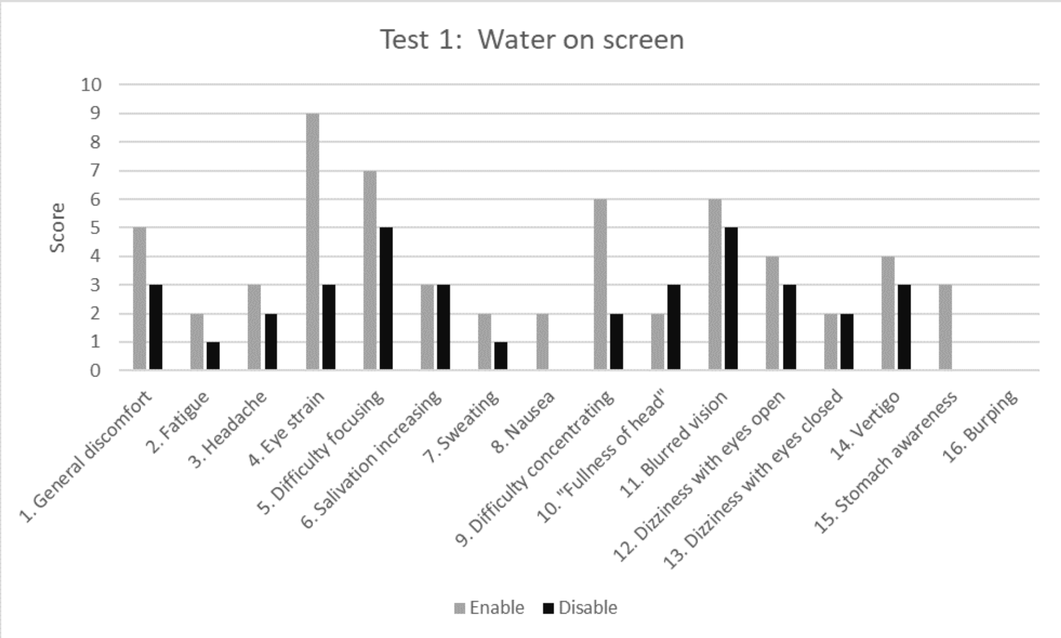

Figure 4.3: The results from Test 1 VR SSQ, with grey being enable and black being disable.

The final total score was found by adding up all total scores from each participant and it was found to be 3131,18 for the Enabled version with an average of 293,83 for each participant and the Disabled version had a final total score of 1986,01 and an average of 180,54. Visualizing the data in a boxplot, Figure 4.4, the whiskers, box, and median were all lower for the Disabled version compared to the Enabled version. There was a single outlier for the Disabled version with a score of over 800, which was higher than the max value of the Enabled version. The lowest score for the Disabled version was on zero where the Enabled version had a lowest score of around 60. Looking at the SSQ data of the outlier they answered *Severe* for both question 5. *Difficulty focusing* and 11. *Blurred vision*. While the participant answered both question 4. *Eye Strain* and 9. *Difficulty concentrating* with *Medium*. Four questions received the answer *A little*, 1. *General discomfort*, 10. *"Fullness of head"*, 12. *Dizziness with eyes open*, and 14. *Vertigo*. The rest of the questions were answered with *None*.

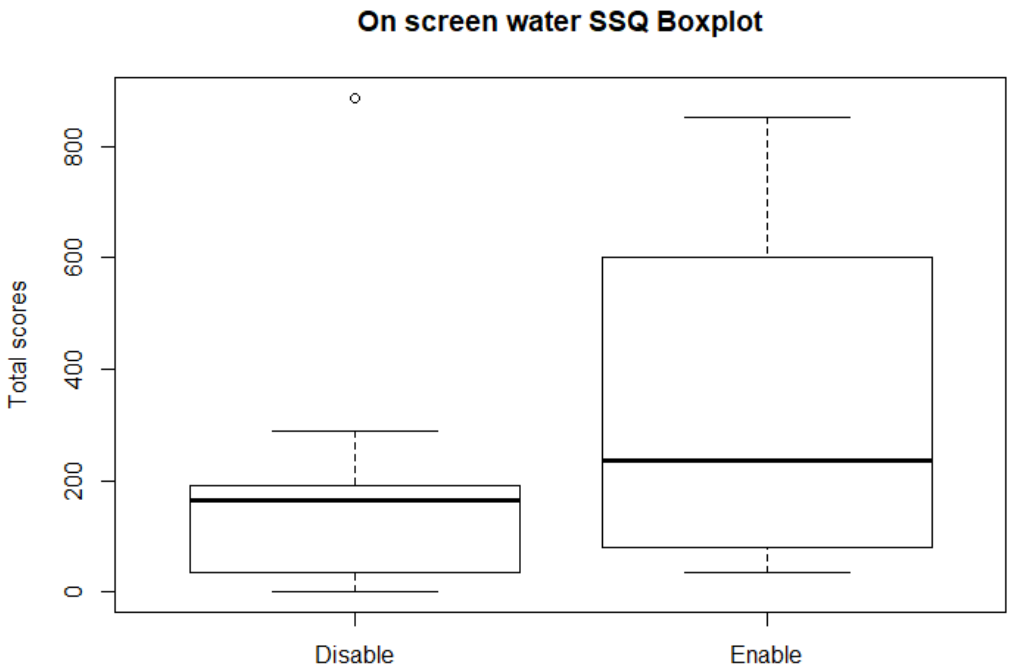

Figure 4.4: Boxplot from Test 1 SSQ, visualising the outcome of the 20 participants 50% of those tested with water effects enabled.

4.1.3 Test 1: Discussion

The scores for each question in the SSQ, Figure 4.3, and the final total scores for the two versions, seems to show a tendency for the Disabled version to cause less VR sickness compared to the Enabled version. When looking at the scores for each question it is the Enabled version that made the participants nauseous or at least making them more aware of their stomach, which is seen as the step before feeling nauseous. Question 4. *Eye strain* and 9. *Difficulty focusing* both has the biggest difference between the two versions. The reason for the Enabled version scoring higher of the two could be due to the participant trying to see either through or around the water drops which could cause straining or difficulties for their eyes. This straining could also have had an effect on why the participants trying the Enabled version scored higher on 2. *Fatigue* as all participants was seated through out the entire testing period. This single point of difference between the two versions could be due to noise of data. It could be the same trend as seen for question 3. *Headache* as well. The final calculated scores of the test show that the Disabled version has the lower scores compared to the Enabled version both for the final total score and the average, except for the single outlier with its score over 800. This would indicate that the participants trying the Disabled version would feel less VR sickness when testing. The outlier could be due to that participant may be very sensitive to VR sickness or motion sickness generally. Though this seems less likely as the participant did not feel nausea or show an increase in stomach awareness. As mentioned the participant answered *severe* for question 5. *Difficulty focusing* and 11. *Blurred vision*, both could be due to a blurring and disturbing effected caused by water drops appearing on the screen. In addition it could also be caused by mismatch of the lens distance.

Briefly mentioned in Section 2.2, VR sickness, immersion has an effect on VR sickness, it could be that the Enabled version by implementing a small part immersion makes the participant reflect on all the other immersion signs missing, such as sound. This reflection could have been done consciously or subconsciously. It could be that adding sounds for the test would change the scoring as the Enabled version as it could be seen as more immersive. The speed could also have had an effect on the score, if participants found it too fast or slow, this could be down to a personal preference. In an effort to lessen the effect this may have had on the differentiation of the two versions both were going the same constant speed. It was observed that under testing that six participants would look behind them, do a 180^o degree turn. This could lead to an increase of VR sickness as the participant could no longer see what was happening in front of them and could there for not predict turns. This did not appear to be the case, when looking at the individual total score. Their previous VR experience ranges from *None to A lot* which makes it seems there looking behind no matter previous VR experience would not increase VR sickness for this test. The results found here may not be the same for different effects, but this test will be used as a constant for the project. From the results it was concluded for later testing that effects such as water drops appearing on the screen of the HMD will not be included.

4.2 Test 2: Teleport

4.2.1 Test 2: Experiment

As mentioned in Section 3.1.1, Teleport to dropoff, two kinds of teleports were suggested. As both methods were closely similar only one would be implemented in the final version of the VRE. In order to find which method were the most suitable one a test was devised. The test had 22 participants, 11 females, with an age span of 21 to 32. The participants would have none to a lot of VR experience, 11 had no experience, five had a little, three answered medium amount of VR experience, and two participants said they had a lot of VR experience.

The procedure of the test had the participants try one of the two teleporting methods three times. For every reiteration the participants were supposed to teleport over different distances, a small, a medium, and a large teleport. The small teleport was to the city, the medium to the water treatment plant and the long to the farm. The duration of the teleport movement was depending on the locomotion method and the place the participant would chose. The linear movement would always take 10 seconds for each target, while the Bézier movement would vary, taking 10 seconds to reach the city and up to 18 to reach the farm. After the VRE test, which took between two to five minutes, the participants were asked to fill out the SSQ. Before putting on the HMD the participant got an introduction on how to activate the teleport method, but not what would happen or how they would move when activated. When inside the VRE the participants were asked to collide with the previously mentioned targets, being city, the water treatment plant, and the farm. When landing in the marked point from the teleport, the participants would be moved back to the start position.

4.2.2 Test 2: Results

The tests results are visualized in Figure 4.5, which display the scores of each of the questions. *General discomfort* where Linear movement has a score of eight, while the Bézier movement had half as much. The same can be seen with the Linear movement having the higher score. This can also be seen in question 7. *Swearing*, 8. *Nausea*, 10. *"Fullness of head"*, 11. *Blurred Vision*, 13. *Dizziness with eyes closed*, 14. *Vertigo*, 15. *Stomach awareness*, and 16. *Burping*. For the questions 2. *Fatigue*, 4. *Eye strain*, and 5. *Difficulty focusing* the two methods has the same score. For every question where Bézier movement has a higher score the difference is only by one point.

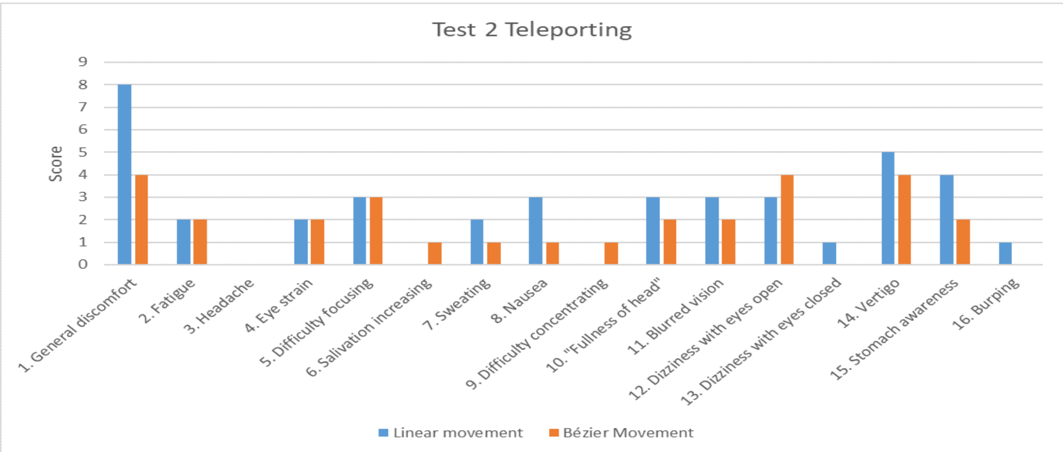

Figure 4.5: The results from Test 2 VR SSQ, with blue being Linear movement and orange being Bézier movement.

Linear movement had 2598,47 as its final total score, with an average of 236,22 per participant, while the Bézier movement had a final score of 1425,83 with an average of 129,62 per participant. Figure 4.6 gives a visual representation over the total score for each participant. From Figure 4.6 it can be seen that the Bézier movement has a lower score for both box and whiskers when compared to the Linear movement.

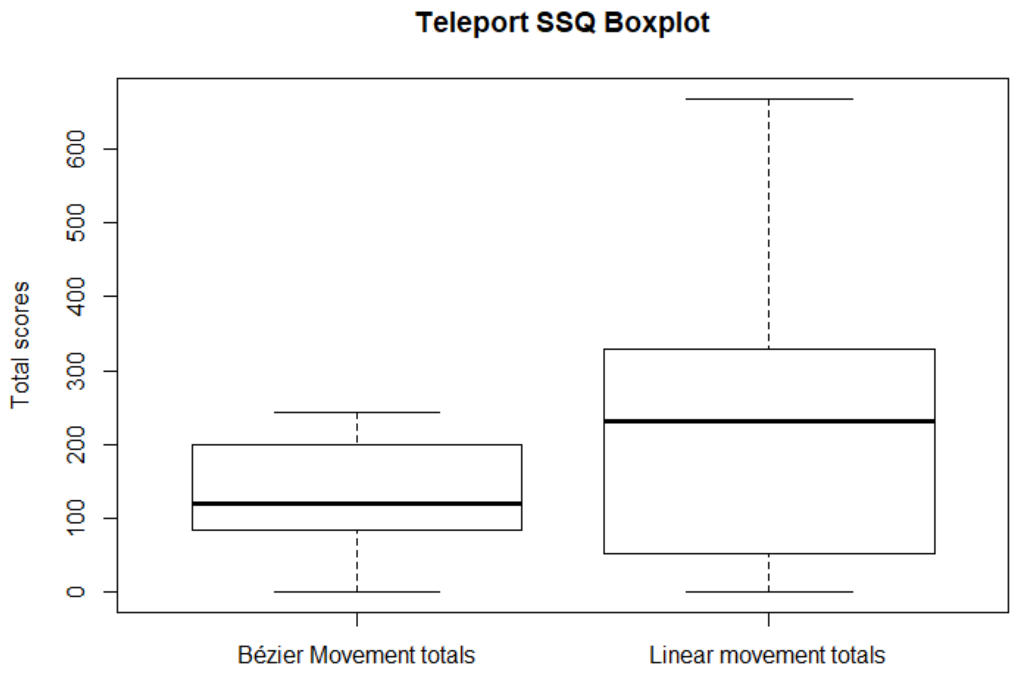

Figure 4.6: Boxplot from Test 2 SSQ, visualizing the outcome of the 22 participants.

4.2.3 Test 2: Discussion

From the results of Test 2 as in Section 4.2.2, Test 2: Results, it can be seen that the Bézier movement had the lowest total score which means it was the locomotion method that had the least amount of VR sickness symptoms. Some of the results may have been affected by the room used for testing, as it was small and difficulty to decrease temperature without having an open door. This open door could have led to distractions for a subset of the participants. This may also have had an effect on sweating. Other than the heat in the room, the HMD may also have had an effect as few to none of the participants adjusted the lenses in the HMD to their specifications, this could have had an affect on 3. *Headache*, 4. *Eye strain*, 5.

Difficulty focusing, 11. *Blurred vision*, and 12. *Dizziness with eyes open*. The lack of adjustment may help with some of the explanation for why 4. *Eye strain* and 5. *Difficulty focusing* have the same score for both locomotion methods. The reason for this absence of adjusting may be due to the lack of VR experience for a lot of the participants, but even those claiming to have a lot of experience were not observed to adjust the lenses. Looking at question 2. *Fatigue* the reason for both methods having the same score could be because both methods are operated the same way, the only difference is how the participant move towards their target. For question 6. *Salivation increasing* and 12. *Dizziness with eyes open* Bézier movement has the higher score, this could be due to the constant graduate decent instead of the sudden drop from Linear movement. For Linear movement this sudden drop may explain the higher score in question 8. *Nausea*, 13. *Dizziness with eyes closed*, 15. *Stomach awareness*, and 16 *Burping*. The boxplot as displayed in Figure 4.6, visualizes the lowest score for each of the locomotion methods, which is zero. This is due to some participant having chosen *None* for each of the 16 questions in the SSQ. Looking at the highest score Linear movement goes above 600, whereas Bézier goes to around 250. This trend seems to follow in for the box in each of the locomotion methods, as Bézier has the smaller box. Though Linear movement has the lower values for its box, the median is still higher than the median of the Bézier. As mentioned in Section 4.2.1, Test 2: Experiment, each participant only tried one of the two locomotion method. The reason for this was to try to avoid bias, which could have an effect on the questionnaire and the results. This also meant that it was not possible to see if problems such as eye strain and blurred vision came from an improper adjusted HMD or if there are no correlation. In almost all 16 questions in the SSQ Linear movement had the highest or an equal score to the Bézier method, it was only in 6. *Salivation increasing*, 9. *Difficulty concentrating*, and 12. *Dizziness with eyes open* where the Bèzier movement had higher values. This seems like a good indication that the Linear method was not a good choice in an effort to reduce VR sickness when compared to the Bézier method. Furthermore as the participant tested for a limited time it can only be assumed that if they were to try it for longer periods the results would only get more severe. All in all it would seem that using the Bézier movement method gave the user the least amount of VR sickness, based on the SSQ total score, and was therefore used for the Teleport method in the VRE.

4.3 Test 3: Controller

4.3.1 Test 3: Experiment

In order to determine which of the two Orientation methods, mentioned in Section 3.1.2, Controller movement, to use for the final prototype, a third test was conducted. The test had 21 participants, seven females, with a total age span of 22-29. They had various previous VR experience, from which seven participants had no previous VR experience, six participants had a little experience, five had medium experience, and three said they had a lot of experience. For this procedure each participant tried both orientation methods. In regards to order, the participants were presented with one of two methods, which would change from participant to participant. Meaning if one started with orientation of the controller the next started with Display orientation and vise versa.

In each of the versions the participants got to try the method for a minimum of three times, as they had to move to the same three targets which was also used for the other tests. The first target was at a close range, next was at a medium range, and the last was at a long range. After having hit each target the participant would be moved back to the start position. When the participant had visited all three locations they were asked if they wanted to keep exploring the VRE. After the participants said they explored enough they were asked to fill out the SSQ. Having answered the 16 questions, the participants got to try the other version of the locomotion method. For the other locomotion method, the participants were again asked to move to the same target, while they afterwards had the possibility to continue testing, before filling out the SSQ. Lastly the participant were asked nine Computer System Usability Questions (CSUQ) that were based on the "Standardized Usability Questionnaires" [2], this questionnaire can be found in appendix B.

4.3.2 Test 3: Results

Starting with the SSQ results which can be seen in Figure 4.7 the two highest scoring results question 1. *General discomfort* and 14. *Vertigo* are both for Display orientation. Out of the 16 different questions the Display orientation had a higher or equal score to the Controller orientation in 12 cases. It was in question 3. *Headache*, 4. *Eye strain*, 12. *Dizziness with eyes open*, and 13. *Dizziness with eyes closed* where Controller orientation has the higher score.

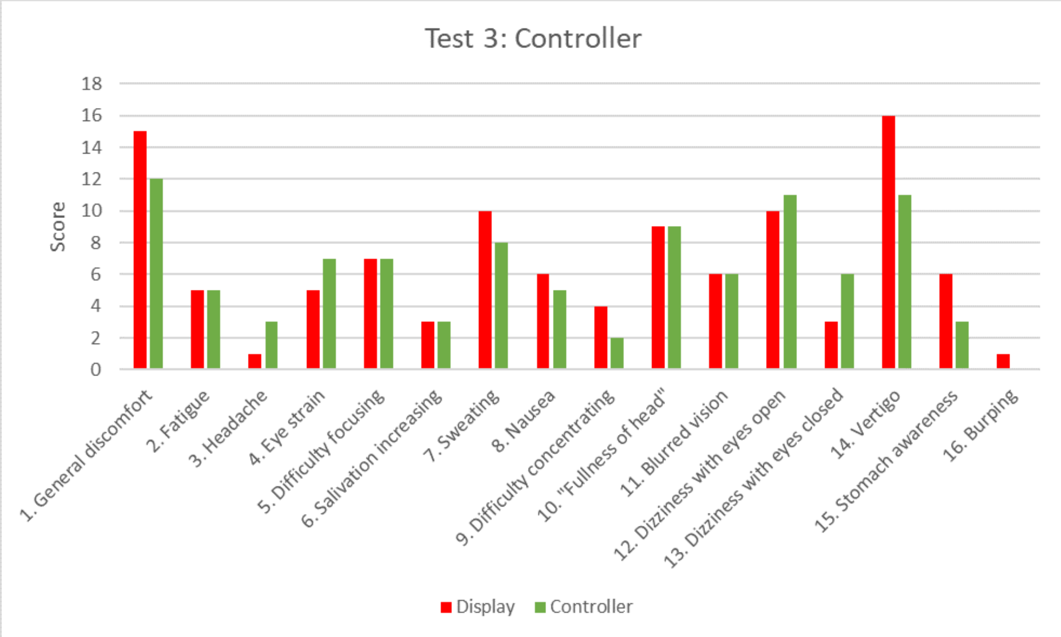

Figure 4.7: The results from Test 3 VR SSQ, with red being Display orientation and green being Controller orientation.

Utilizing the data generated from the SSQ resulted in the final score of 5704,32 in case of the Display orientation with an average of 271,63 while the Controller orientation had a total of 5267,11 and an average of 250,81. Figure 4.8 was made of the total scores form each of the participants. Even thought the Display orientation method has both the highest total score and average, compared to the Controller orientation method, the box is generally lower than the controllers box plot. When looking at the maximum and minimum values of both orientation methods it can be seen that the lowest value is set equal to zero, while the maximum value of the Display had a higher result. There is a single outlier which can be seen under the Display orientation with an value of over 1000. For the outlier they answered *Severe* in question 11. *Blurred vision* and 14. *Vertigo*. For question 5. *Difficulty focusing*, 7. *Sweating*, and 12. *Dizziness with eyes open* the answers were *Moderate*. They answered *Slight* in question 1. *General discomfort*, 4. *Eye strain*, 9. *Difficulty concentrating*, 10. *"Fullness of head"*, 13. *Dizziness with eyes closed*, and 15. *Stomach awareness*. For the last questions, 2. *Fatigue*, 3. *Headache*, 6. *Salivation increasing*, 8. *Nausea*, and 16. *Burping* were given a score of *None*.

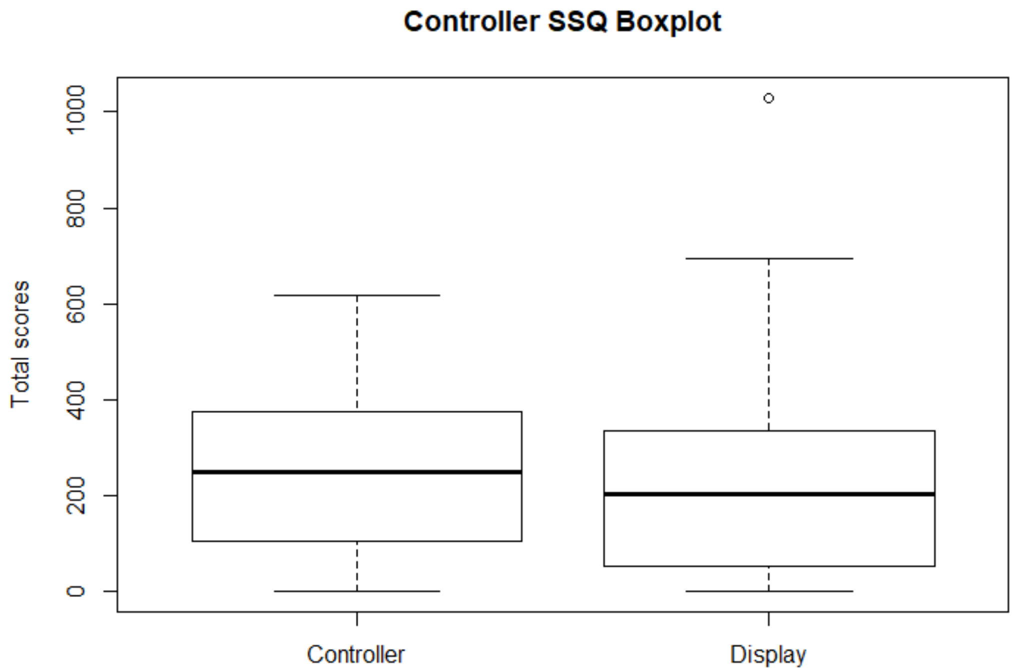

Figure 4.8: Boxplot from Test 3 SSQ, visualizing the outcome of the 21 participants.

Visualising the results of the CSUQ, the distribution for each participant on each of the questions can be seen on Figure 4.9 to Figure 4.17, the full table of answers can be seen on Appendix C. To calculate the values gotten from the CSUQ a unpaired-, two-tailed-, within-group- Mann-Whitney-Wilcoxon calculation was performed which gave a p-value of 0,1489.

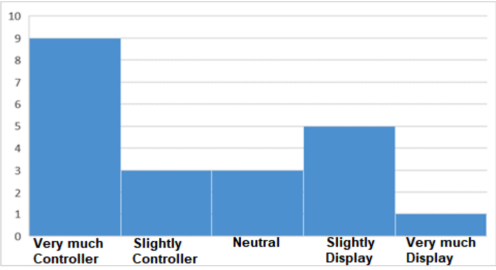

Figure 4.9: The Histogram for the CSUQ Question 1, Which method was easier to use?

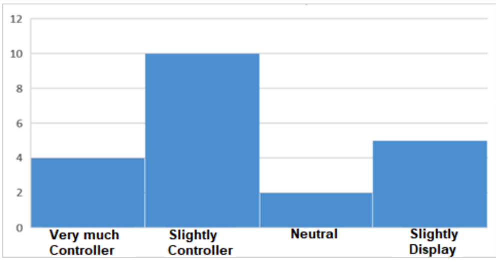

Figure 4.10: The Histogram for the CSUQ Question 2, Which method simpler to use?

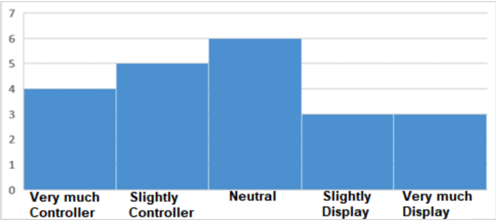

Figure 4.11: The Histogram for the CSUQ Question 3, Which method made it more effective completing the task?

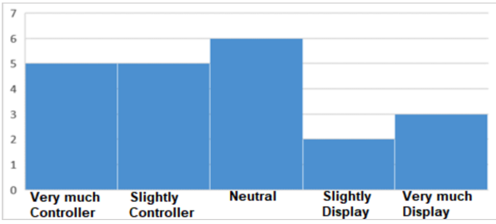

Figure 4.12: The Histogram for the CSUQ Question 4, Which method made it quicker to complete the task?

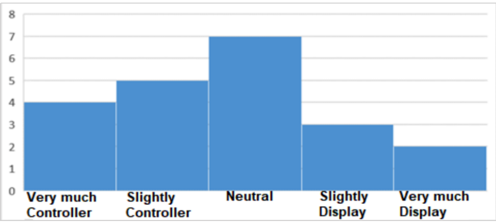

Figure 4.13: The Histogram for the CSUQ Question 5, Which method made it more efficient to complete the task?

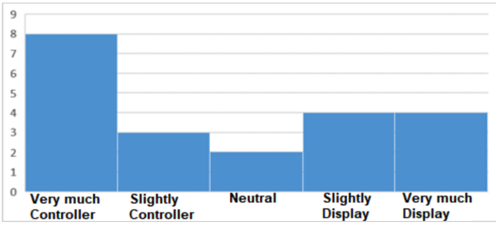

Figure 4.14: The Histogram for the CSUQ Question 6, Which method was most comfortable to use?

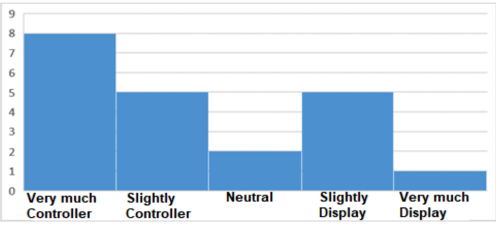

Figure 4.15: The Histogram for the CSUQ Question 7, Which method was easier to learn to use?

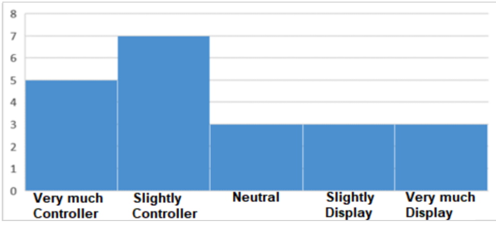

Figure 4.16: The Histogram for the CSUQ Question 8, Which method behaved most like you expected?

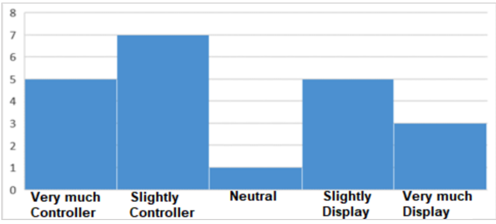

Figure 4.17: The Histogram for the CSUQ Question 9, Which method was overall most satisfying?

4.3.3 Test 3: Discussion

The SSQ results for the Controller orientation method had the lowest total score and average, meaning the participants felt this method gave the least amount of VR sickness. Though it has the lowest total score it had a higher box placement in Figure 4.8 compared to Display orientation. The reason for the Controller orientation having a lower total score and average but a higher box placement in Figure 4.8 could be due to the outlier, which would drive up the total score and average for Display orientation. For the outlier their high total score mainly originates from question 11. *Blurred vision* and 14. *Vertigo*, which was answered with *Severe*. As their tried the Display orientation method, this could be due to them moving their head rapidly around to look around the VRE or try to find the target to hit. It would seem that having to turn around themselves, looking up and down may have caused the participants lose their orientation of the real world which could cause answering *Severe* in 14. *Vertigo*. A reason for the participant scoring high in 11. *Blurred vision* could be due to the position of the lenses not being adjusted to the participant specifications. Though being the highest total scoring participant in this test it would seem that the Display method did not cause a feeling of nausea, though if the test would have continued longer they might as they got an increased stomach awareness while testing. Looking at the whiskers both methods have zero as the lowest score, due to participants not feeling any VR sickness symptoms, but the Controller has a lower higher whisker compared to the Display. The tendency for Display to have the higher score in SSQ seems to also be the case when looking at the score for each question in the SSQ, Figure 4.7. As mentioned in Section 4.3.2, Test 3: Results, the Display orientation method has the highest or equal score in 12 out of 16 questions. Looking at questions 3. *Headache* and 12. *Dizziness with eyes open* the Controller orientation method has the highest score this could be due to this method giving the participants the ability to walk sideways, strafing, which could lead to an increased feeling of VR sickness.

The CSUQ was calculated using an unpaired-, two-tailed-, within-group- Mann-Whitney-Wilcoxon test which was done because the two methods was not dependent on each other, it was not known which of the two methods was preferred, and each participant tried both methods. The Mann-Whitney-Wilcoxon test form was chosen because of the use of likert scale when the participant had to compare the two orientation methods in the CSUQ test. From the test a significance level was set to 0,05 and a p-value of 0,1489 was found, which meant that there are no significant difference between the two methods leading to the null hypothesis "The two methods are similar" was kept and the alternative hypothesis "The two methods are not similar" could be rejected. The reliability of the CSUQ may have suffered as the test took place over several days with breaks in between. This could have created a difference in how the methods were described to the participants, but an effort was made to repeat the same description each time. In the procedure for this test every second participant started with the same movement method, the reason for this was to avoid bias and to try to avoid side effects carrying over and interfering with the results. If one of the methods was significantly more prone to increase VR sickness and that method was the first for each participant, it may have effected the second orientation method tested afterwards.

The results found in this test may not show a whole picture, as VR sickness may not set in right away but take longer time to appear than the amount the participant had in the test. Though there were not a significant difference between the two methods, based on the CSUQ, there still seemed to be a slight tendency of preference for the Controller method of orientation. This is based on the results from the SSQ and looking at the histogram Figures 4.9 to 4.17 which all seems to favor the Controller method. All results leads to the use of the Controller orientation method for this locomotion method.

4.4 Test 4: Trackpad

4.4.1 Test 4: Experiment

To determine which orientation version to use for the Trackpad movement method, a test was conducted where the participants got to orientate themselves in the VRE based on either the VR controller or the HMD. 22 participants tested, nine of which were female, with an age span of 19 to 35. Of the 22 participants ten had no previous VR experience, seven had a little, two expressed that they had medium experience, and three said they had a lot of experience with VR. Similar to the other tests, the orientation method was changed from one participant to the next, this behavior was chosen to eradicate influence which could be carried over from one method to the other.

Following the same protocol which was used in Test 3, a single participant would test the VRE at the time. The participants received an explanation of how the controllers worked. If the user had further questions the facilitator tried to expand the information given, to such a level where the participant said they understood it. When inside the VRE the participant was asked to move towards a target. Once the target was hit the participant was reset to the start position and a new target was appointed. Each participant was asked to hit the City, the Water treatment plant, and the Farm. When all three required targets were met, the participant were asked to remove the HMD and to fill out the SSQ. This process would then repeat for the second orientation method. Having answered the same questionnaire a second time for the second orientation method, the participants were given the CSUQ, where they were asked to compare the two methods based on various factors.

4.4.2 Test 4: Results

Looking at Figure 4.18, the biggest difference came in the questions 4. *Eye strain*, 9. *Difficulty concentrating*, and 12. *Dizziness with eyes open*, where there were a difference of three points for each question. The second highest differences were two points which can be seen in question 1. *General discomfort*, 5. *Difficulty focusing*, 10. *"Fullness of head"*, and 11. *Blurred vision*. Only burping was given a score of zero.

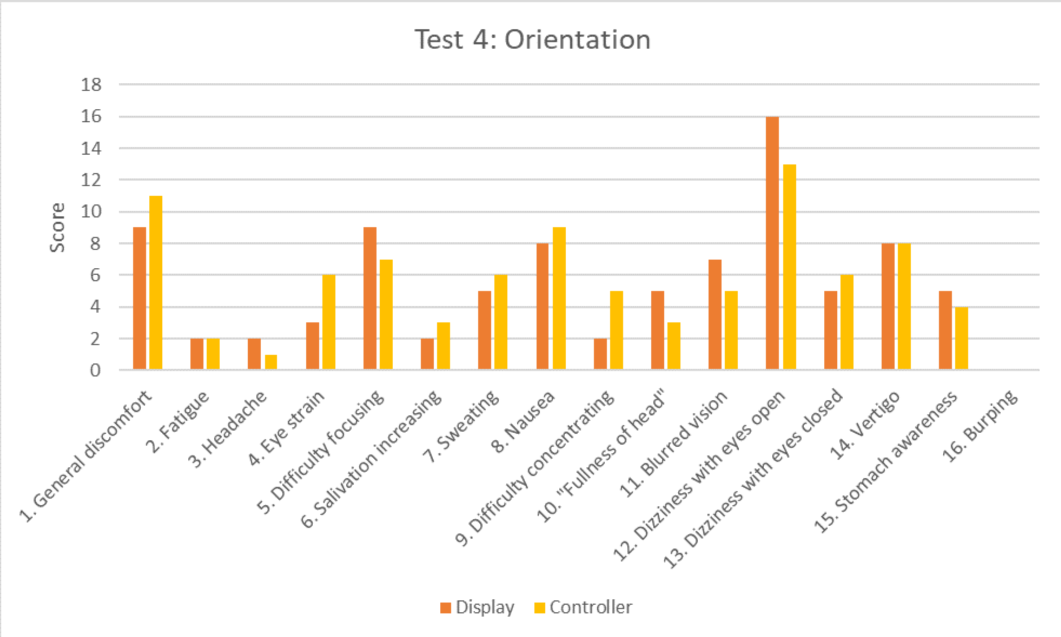

Figure 4.18: The results from Test 4 VR SSQ, with orange being Display orientation and yellow being Controller orientation.

Calculating the final total score for the Controller orientation method led to a total of 4972,11 with an average of 236,77 whereas the Display method had a total score of 5125,15 with an average of 244,10. Figure 4.19 was based on the total score from each participant, where the Controller method in general had the overall lower score, except for the median which was lower in Display. Both methods had a single outlier each. For the Controller method the lowest value was also part of the box. By calculating the p-value, the same way as with Test 3, it was found to be 0,9301.

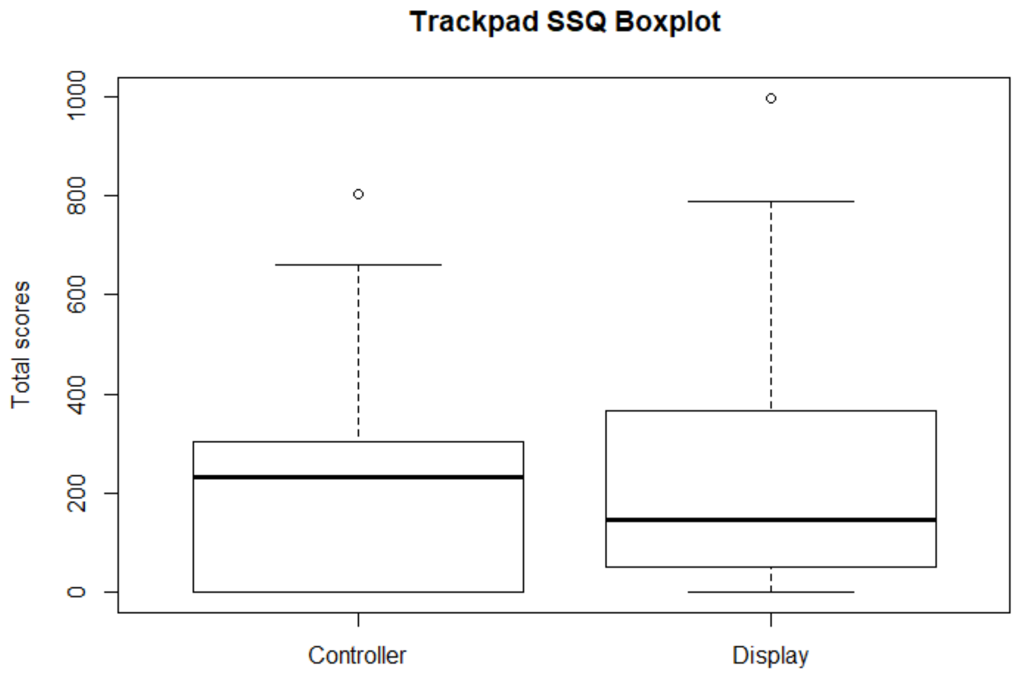

Figure 4.19: Boxplot from Test 4 SSQ, visualising the outcome of the 22 participants.

Looking at the results from the CSUQ the Controller method had a total of 88 votes and the Display method had a total of 81 votes, these are from counting both the *slightly* and *very much* option for both of the methods. Out of the 88 votes for the Controller 35 was *slightly Controller* and 53 *very much Controller*. Out of the Displays 81 votes the split was 52 for *slightly Display* and 29 for *very much Display*. This left 29 votes for *neutral* out of the possible 198 votes. All data is visualized in Figure 4.20 to Figure 4.28. The full table of answers can be seen on appendix D.

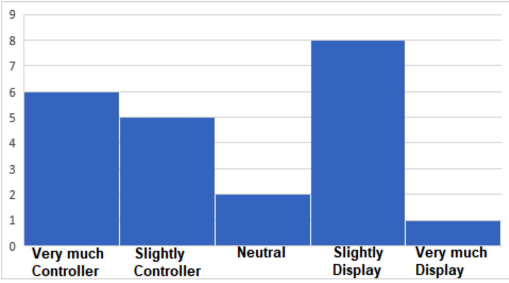

Figure 4.20: The Histogram for the CSUQ Question 1, Which method was easier to use?

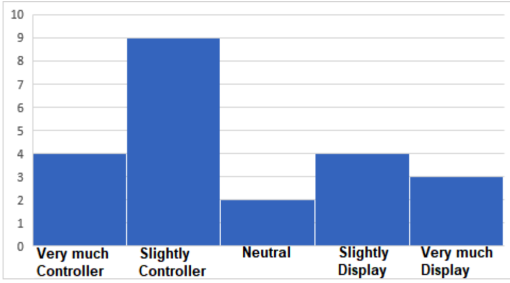

Figure 4.21: The Histogram for the CSUQ Question 2, Which method simpler to use?

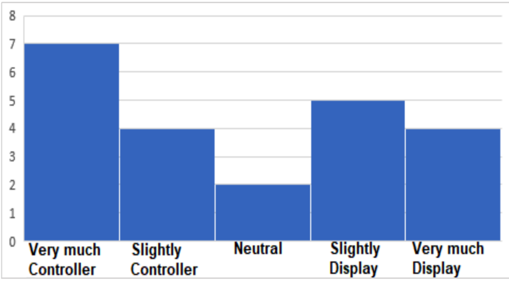

Figure 4.22: The Histogram for the CSUQ Question 3, Which method made it more effective completing the task?

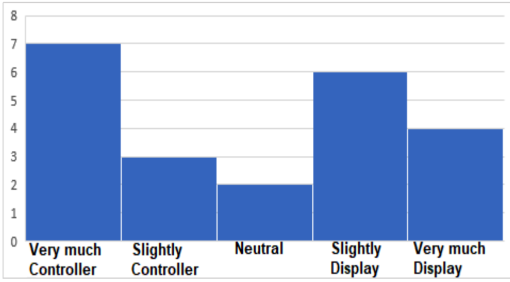

Figure 4.23: The Histogram for the CSUQ Question 4, Which method made it quicker to complete the task?

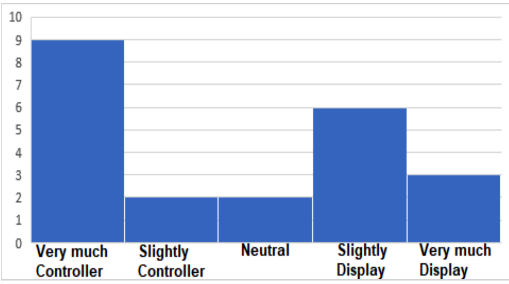

Figure 4.24: The Histogram for the CSUQ Question 5, Which method made it more efficient to complete the task?

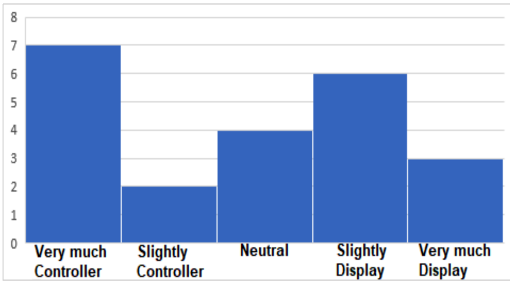

Figure 4.25: The Histogram for the CSUQ Question 6, Which method was most comfortable to use?

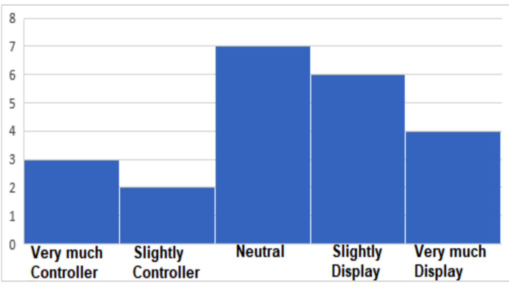

Figure 4.26: The Histogram for the CSUQ Question 7, Which method was easier to learn to use?

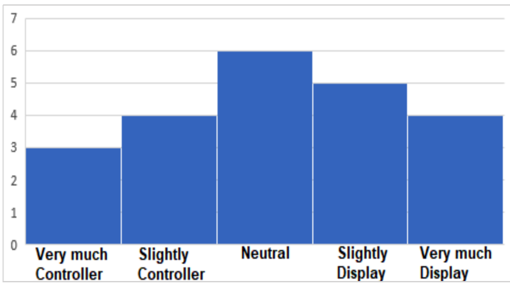

Figure 4.27: The Histogram for the CSUQ Question 8, Which method behaved most like expected?

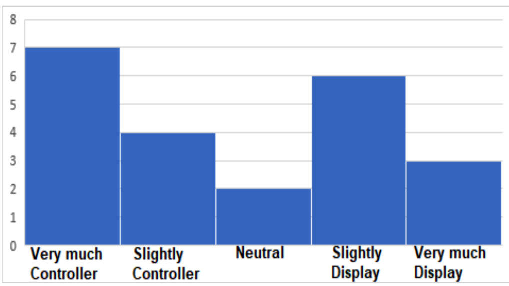

Figure 4.28: The Histogram for the CSUQ Question 9, Which method was overall most satisfying?

4.4.3 Test 4: Discussion

From the results of the SSQ, both the bar plot Figure 4.18 and the boxplot Figure 4.19, seem to show a tendency of the two being similar. There is only a difference of seven votes between the two different methods of orientation, though the Controller has more *Very much* votes compared to the Display method. This tendency of preferring the Controller is also visible for the final total score for the SSQ with the Controller having a 4972,11 and an average of 236,77 per participant and the Display having 5125,15 and an average of 244,10. This gives a very little difference in total score and average between the two methods. Looking at Figure 4.19 the box for the Controller is generally lower than the box from the Display, the same goes for the whiskers with the Controllers lowest whisker being same as the lowest part of the box. Only the median for the Controller is higher than the median of the Display, which means that there seems to be a larger amount of higher numbers for the controller but the few very low once, the zeros, are pulling the average down lower than the average of the Display. From the CSUQ the p-value was found to be 0,9301 which is way higher than the set significance value set to 0,05. From this the null hypotheses can be kept and the alternative hypotheses rejected. Though having a p-value of 0,9301 may indicate that there were either a lot of noise and/or variation in the data or maybe there were too few data samples. Which could both be the case for the test, without testing on more participants. It could also be the case that the participants found no real difference in the two orientation methods, leading to the null hypothesis being true.

Like the other test before this one, in an effort to avoid bias the participants started with alternating locomotion methods, meaning every other participant started with the same method. It was also done in an effort to avoid side effects carrying over from one method to the other and affecting the questionnaire. In an effort to get more participants for the test it was done over two days, with three days in between. This may have affected the reliability as the description of the methods may have changed, though an effort was made to use the same explanation for each participant. While testing a few observations were made. One of the participants would place both thumbs on the trackpad. This resulted in either a full stop or unpredictable movement, which could have had an affected the two questionnaires. It is believed this happened because of the participant being nervous over the first time trying VR. It was also observed that under both locomotion methods participants would mostly use one axis on the trackpad, the y-axis as seen on Figure 3.8. This could be due to strafing could cause the participants discomfort or they may not have found it useful as the targets they were asked to hit could all be reached moving in a straight line from the starting position. Though there are no significant difference in the CSUQ and no big difference in the final SSQ score between the two locomotion methods, it was decided to go with the Controller orientation method. This due to higher amount of votes in the CSUQ with 88 for

Controller compared to the 81 for Display, where most of the 88 votes were in favor of *very much Controller*. This method also had the lowest final total score from SSQ. As there are no big difference in the two methods choosing Controller may suffer for the final test as participants could have an emotional preference which could result in discomfort using a different orientation method.

4.5 Test 5: Final Test

4.5.1 Test 5: Experiment

The Final test had the participants tested individually also seen in the other tests. The participants got an introduction to the focus of the project and why it was relevant to research how to fall in a VRE. Initially a questioner regarding age, gender, tendency towards motion sickness, tendency to fear of heights, and the previous experience with VR was provided. This questionnaire can be seen in the appendix A and E. For the final test a scheme was made in order to test each combination of locomotion methods once, see appendix G for the full table. This was done to ensure that the result data would not be influenced by the order of which the locomotion methods were tested in. If each combination of all four locomotion method should be tested it would require a minimum of 24 test. Thereby 24 participants were required, of which five were female. the participants had an age span of 21 to 28. When asked about their previous VR experience five answered *Never*, seven *Less than once every three months*, another five answered *Once every three months*, five more answered *A couple times a month*, and lastly two answered *Once a week*. For being scared of heights seven participants answered *Not at all*, 11 participants said *A little*, and six chose *Medium*. Lastly to see the participants tendency for motion sickness eight answered *None*, 13 participants chose *A little* for their answer, two participant felt *Medium* tendency, and a single participant said *A lot*. The participant received an introduction to the specified locomotion method, in case the participant was new to the HMD they got an introduction to VR and the HMD as well. The locomotion methods were tested by defining the same target which the participant had to hit. The target chosen for this test was the Waste water treatment plant, which was also used in the previous test. This target was placed around the middle of the VRE and would provide the participants a chance to look around the virtual world before landing. After this requirement was met, a SSQ was provided for the participant, in the case were a user was unsure additional information regarding the SSQ question was provided. After each of the SSQ's the participant got introduced to the next locomotion method. After all four methods were tested, the participant received a CSUQ comparing the individual methods. When finishing the CSUQ the participant was exposed to one of the six additional scenes which were described in Section 3.2, VR Environment. Upon completion the participant received a final SSQ that had to be answered in regards to the scene. This procedure led to a test with both within-group from having tested each locomotion method and between-group for only testing one of the six scenes per participant. The *C#* code of the four methods which got tested can be found in the appendix I.

As mentioned previously in this Section, for all combination of locomotion to be done once 24 participants were needed. This means that each of the six scenes created would not be tested by each combination of the locomotion methods. The reason for this is that it would require 144 different participants and it was deemed to out of the scope of this project.

4.5.2 Test 5: Results

The first SSQ regarding Method 1 - Teleport, had a total value of 1331, 2 points, with an average of 55, 47 points per participant. The median value for the first SSQ was calculated to be 34, 83. Out of the 24 participants a total of six answered with zero points for the entire SSQ while the highest value was 213, 36.

For the Second SSQ for Method 2 - Trackpad, had a total value of 1091, 64 points, with an average of 45, 49 points per participant. The median value for this SSQ was calculated to be 20, 29. In this SSQ total of four participants answered with zero points while the highest value was 206, 96.

The SSQ of Method 3 - Controller, had a total value of 657, 4 points, with an average of 27, 39 points per participant. The median value of this SSQ was 13, 92, while a total of nine answered with zero points, at the same time the highest value was 181, 08.

The SSQ regarding Method 4 - AutoMove, had a total value of 1279, 44 and an average of 53, 31 points per participant. The median was calculated to be 27, 84 and a total of four answered with zero points for this SSQ, the highest value was 181, 08 as in the third method, however from a different participant than the one from method 3, Controller. All locomotion methods can be compared using a boxplot as displayed in Figure 4.29.

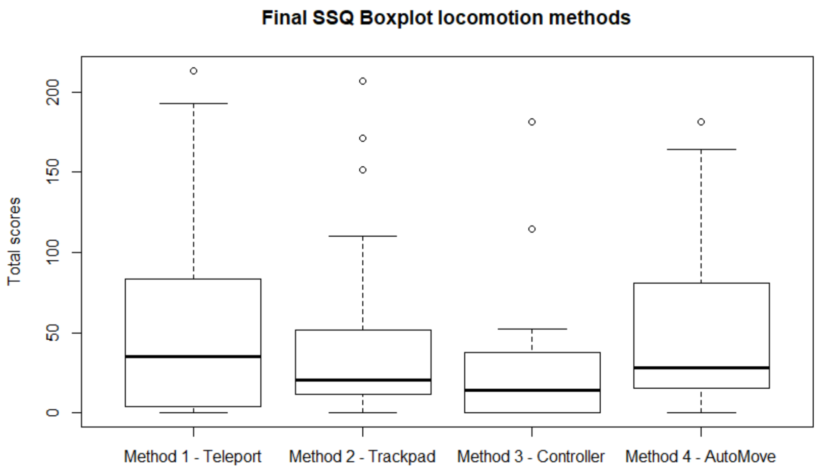

Figure 4.29: A boxplot graph comparing all the 4 locomotion methods tested. The circles indicate outliers.

For the CSUQ the 24 participants replied with a total of 216 points, out of which five were not given to any movement method. The Controller hereby received the highest total score of 73, followed by the Automove method with 65. The Teleport received 45, while the Trackpad had the fewest total amount of points with 28. For this CSUQ the template was altered to an extend that the users had to select one of the nine possible responses, while not grading each method individually. The outcome of the CSUQ can be seen in Figure 4.30 to 4.38.

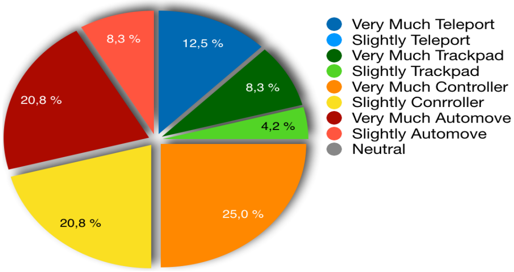

Figure 4.30: The pie chart for the final CSUQ Question 1, Which method was the easiest to use?

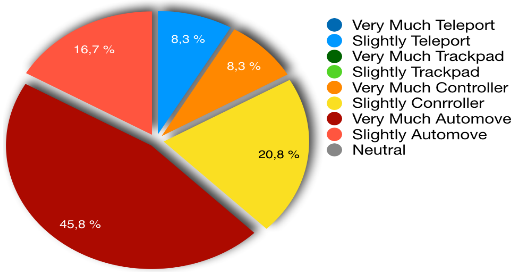

Figure 4.31: The pie chart for the final CSUQ Question 2, Which method was simplest to use?

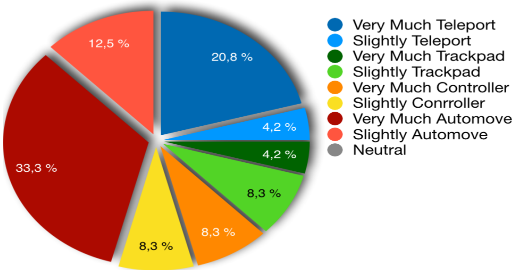

Figure 4.32: The pie chart for the final CSUQ Question 3, Which method was most effective in completing the task?

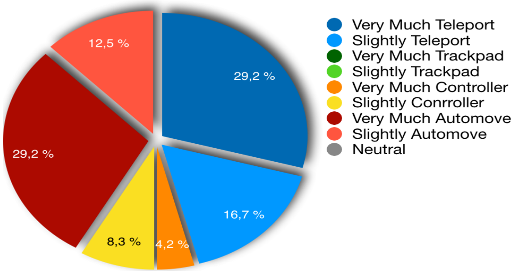

Figure 4.33: The pie chart for the final CSUQ Question 4, Which method was the quickest to complete the task?

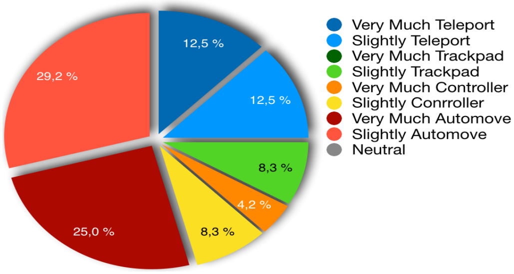

Figure 4.34: The pie chart for the final CSUQ Question 5, Which method was the most efficient to complete the task?

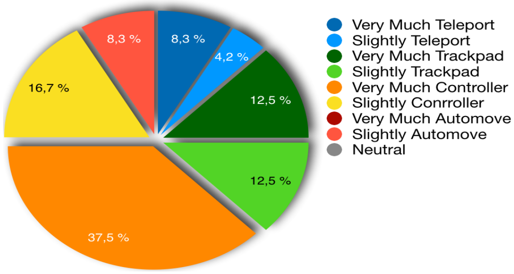

Figure 4.35: The pie chart for the final CSUQ Question 6, Which method was the most comfortable to use?

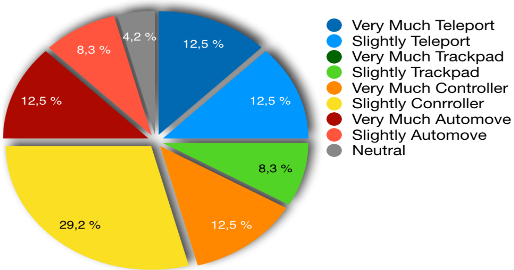

Figure 4.36: The pie chart for the final CSUQ Question 7, Which method was the easiest to use?

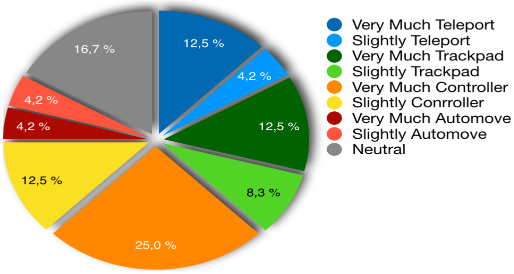

Figure 4.37: The pie chart for the final CSUQ Question 8, Which method was the easiest to learn to use?

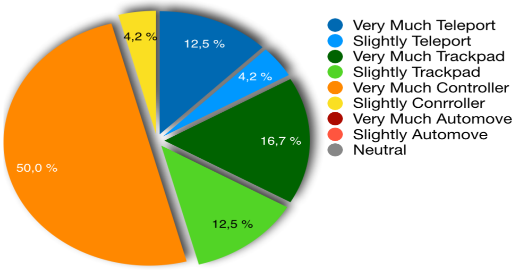

Figure 4.38: The pie chart for the final CSUQ Question 9, Which method was the overall most satisfying?

The final SSQ regarding the scenes and not the locomotion methods had all the 24 participants testing one individual scene. Thereby four participants tested each scene. The results regarding: total score, mean, median and max value can be found in Table 4.1. The same data can be seen visualized in a boxplot in Figure 4.39.

Scene Names	Total	Mean	Median	Max Value
Scene: City	243.94	60.985	43.39	157.2
Scene: Dessert	248.32	62.08	75.41	97.5
Scene: Farm	212.18	53.045	46.56	119.1
Scene: Mountain	169.58	42.395	47.15	75.3
Scene: River	291.32	72.83	58.29	174.7
Scene: WaterTreatment	375.62	93.905	78.81	204.1

Table 4.1: The test results of the individual scenes, each scene was tested four times.

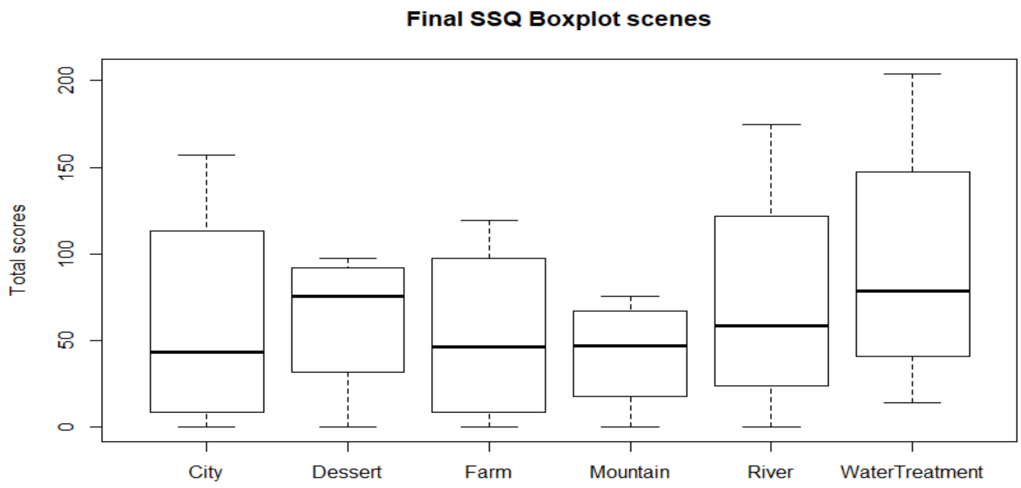

Figure 4.39: The boxplot of the individual scenes, visualizing the data outcome of the SSQ given after the test of the scene, with 4 data points per scene.

Figure 4.40 to 4.48 are tables each holding the p-value found comparing each locomotion method in each of the nine CSUQ questions. Each p-value was found using a one-tailed Mann-Whitney-Wilcoxen calculation.

Q1	Teleport	Controller	Trackpad	Automove
Teleport		0,02301	0,9713	0,1941
Controller	0,02301		0,01582	0,3261
Trackpad	0,9713	0,01582		0,1619
Automove	0,1941	0,3261	0,1619	

Figure 4.40: Table of the p-values for Question 1. Which method was the easiest to use?

Q2	Teleport	Controller	Trackpad	Automove
Teleport		0,06171	0,1616	5,02E-05
Controller	0,06171		0,004973	0,006567
Trackpad	0,1616	0,004973		5,45E-06
Automove	5,02E-05	0,006567	5,45E-06	

Figure 4.41: Table of the p-values for Question 2. Which method was the simplest to use?

Q3	Teleport	Controller	Trackpad	Automove
Teleport		0,4229	0,2248	0,1707
Controller	0,4229		0,6743	0,02525
Trackpad	0,2248	0,6743		0,008417
Automove	0,1707	0,02525	0,008417	

Figure 4.42: Table of the p-values for Question 3. Which method was most effective in completing the task?

Q4	Teleport	Controller	Trackpad	Automove
Teleport		0,009492	0,00022	0,8532
Controller	0,009492		0,08103	0,01848
Trackpad	0,00022	0,08103		0,0004972
Automove	0,8532	0,01848	0,0004972	

Figure 4.43: Table of the p-values for Question 4. Which method was the quickest to complete the task?

Q5	Teleport	Controller	Trackpad	Automove
Teleport		0,2617	0,1082	0,05309
Controller	0,2617		0,6266	0,002536
Trackpad	0,1082	0,6266		0,0005165
Automove	0,05309	0,002536	0,0005165	

Figure 4.44: Table of the p-values for Question 5. Which method was the most efficient to complete the task?

Q6	Teleport	Controller	Trackpad	Automove
Teleport		0,002911	0,3024	0,5994
Controller	0,002911		0,03149	0,0003884
Trackpad	0,3024	0,03149		0,1082
Automove	0,5994	0,0003884	0,1082	

Figure 4.45: Table of the p-values for Question 6. Which method was the most comfortable to use?

Q7	Teleport	Controller	Trackpad	Automove
Teleport		0,304	0,1082	0,7788
Controller	0,304		0,007219	0,1892
Trackpad	0,1082	0,007219		0,1955
Automove	0,7788	0,1892	0,1955	

Figure 4.46: Table of the p-values for Question 7. Which method was the easiest to learn to use?

Q8	Teleport	Controller	Trackpad	Automove
Teleport		0,1243	0,7615	0,3789
Controller	0,1243		0,2056	0,01693
Trackpad	0,7615	0,2056		0,2261
Automove	0,3789	0,01693	0,2261	

Figure 4.47: Table of the p-values for Question 8. Which method behaved the most like expected?

Q9	Teleport	Controller	Trackpad	Automove
Teleport		0,005787	0,3538	0,04095
Controller	0,005787		0,03903	3,33E-05
Trackpad	0,3538	0,03903		0,004995
Automove	0,04095	3,33E-05	0,004995	

Figure 4.48: Table of the p-values for Question 9. Which method was overall the most satisfying?

4.5.3 Test 5: Discussion

Some of the VR sickness symptoms might have been appearing for the participants later on during that day after having tested, this was however not followed up on.

Locomotion methods

Regarding the CSUQ the questions 2 to 5 might have been disadvantageous for other methods than Automove, since it scored above 40% in those questions as seen in 4.31 to 4.34. The reason for this high score could be due to the way the locomotion method works as it does not require the participant to do anything. It would translate them to the target in a set speed and path, leading to no learning on the participant's side required. With no time used or needed to learn the Automove method it can be easy to see why it would get high scores as simplest to use, question 2. Question 3 asked which method was the most effective, question 4 asked for the quickest method to complete the given task, and question 5 asked to the most efficient method, the reason for the Automove method to get such a high score in these questions as well could be due to it moving in a constant speed straight towards the target. Where other methods either allowed the participant to go on detours or moved in a curved motion the Automove method went in a straight line with no way of altering the path. It is this movement that could lead to the participant feeling it was more effective, quicker, and efficient. Multiple participants were also noted to ask if they were allowed to pick the Automove method for the CSUQ, as it seemed like the clear choice. A single participant commented *"The Automove was the simplest and easiest to use (because I didn't need to do anything) but I did not like it, because it just took me somewhere and I had no control over speed or my direction of approach"* which shows that though it is easier to use, it may not be the right choice for the VRE as it takes control away from the participants.

Some of the comments that the participants provided was that a general speed was too fast. This was seen from some participants which wrote or said that some of the different, or all, locomotion methods were too fast. It was after the test recognized that the movement methods did not have the exact same range of speeds, though an effort had been made to match the different speeds, meaning that the Teleport and the Automove method could move the player faster than Trackpad or Controller could. This varying speed between the locomotion methods may have increased VR sickness if the participant had prepared themselves for a certain speed and then it went faster. For participant 11 a problem occurred where the VRE began to move almost twice as fast as was intended for the locomotion methods and the scene at the end. This could cause said participant to have an increase in VR sickness symptoms, even more than what would normally have occurred or when compared to other participants. Looking at participant 11 answers for the

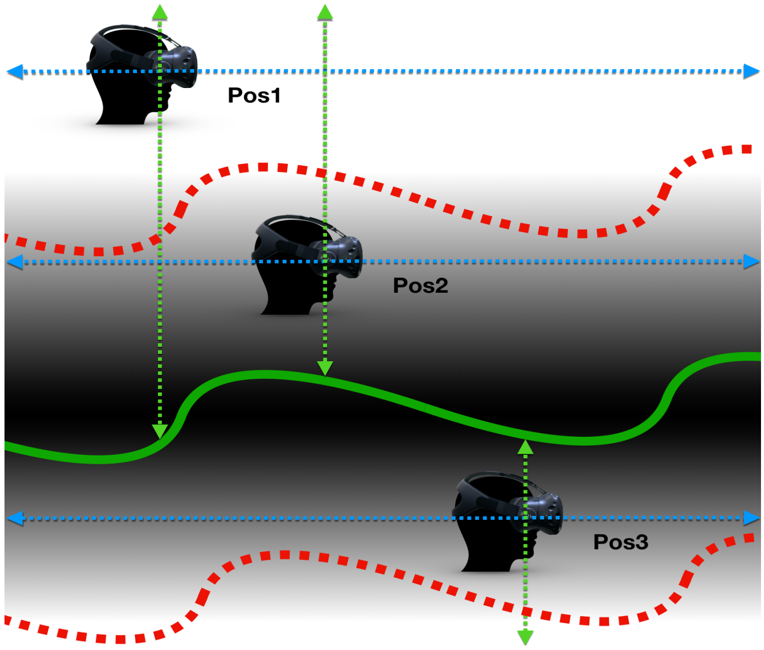

Figure 4.49: The solid green line indicates the surface of the terrain. The red dotted line, the threshold values that have to be surpassed in order to affect the black textures alpha channel. The blue and green dotted lines indicate four of the six measurement directions.

various questionnaires it is unknown how big an effect this may have had. From the different scores, participant 11 is generally in the low end of the data set, but without the speed problem the participant may have had an even lower score.

An issue arose with the FadeOut system, since the participant still was able to move while the screen was pitch black, thus being able to move through walls, which then again enabled the reduction of the blacks alpha value. For a visualization of this issue see Figure 4.49. Meaning the screen would return to normal when behind or below the ground. In addition the "fadeout" gameobject in Unity would be destroyed together with the player before loading the next scene, which also would have been a possibility to see different images being rendered before transitioning to the next scene. While the Teleport, Controller, and Trackpad methods could potentially have had those issues, the Automove method always displayed the issue with the fadeout. This issue may have increased a sense of VR sickness with various participant and may have increased the SSQ score for the locomotion methods where it happened. It could also have caused annoyance for the participants with the quick unfading of the screen and then fading again. Another issue that arose was with participant 3 who experienced controller tracking issue multiple times, which can arise when the controllers are hidden from the base-stations. For other participants it happened rarely, however on participant 3 it was frequent disconnects. This could have lead to frustrations which could have lowered the

CSUQ score for one or more of the locomotion methods. The lose of controller tracking could also lead to unwanted or uncontrollable movement for the participant, e.g. if the participant were using the Teleport method and the controller would lose tracking the participant may be teleported to a different target than intended or start teleporting before the release of the button.

The indicator globe in the Teleport method was never viewed from the inside, this could be because when the movement method got introduced it was mentioned to use the right hand, leading to some participants removing the left controller from the FOV. In removing the left controller from view and thereby the indicator orb a feedforward method was removed. It could be that participants found that this was not needed given that a text would appear over the teleport marker explaining what would be activated given the current landing point. They would also get information of their translation path from the path makers at their feet. These two methods of feedforward information could be enough and adding a third may have been unnecessary. Though this behaviour may only have occurred as all participants knew what target they were going to. Allowing the participants to explore each landmark freely in the VRE may show a different tendency, as the orb would show what the area look like and it would change dependent on landmark indication the given area as a landmark. This is not certain as it may be too much information for the participant and they would block some of it out, leading to the same results see in the testing. A few participants would express discomfort, with either comments at the end of the questionnaire or by saying it out loud, for the last part of the Teleport translation curve. They felt as though it was a straight drop down, which was commented to be uncomfortable. This may give an idea of why the box plot for the Teleport method, Figure 4.29, are the highest scoring and appear to be one of the most varying in the participants scoring of the SSQ.

For the Trackpad method it was observed that participants would use both axis on the trackpad of the controller, which was not observed in Test 4: Trackpad. Other than both axis the same participants would also use the option to slow down or go faster, by moving their thumb closer or further away from the rim of the trackpad. The participants using all the features of the Trackpad method were observed to do it doing landing, in the middle of moving to the target, and before starting to move towards the target. A single participants ended up wobbling after having tested the methods ability to move sideways, which could indicate either a too fast transaction from one movement direction to another or maybe the participant were not prepared for this change in movement direction.

The Controller locomotion method had participants that did not seem to find it intuitive to point the controller straight downwards when they want to descend. One participant would make a zigzag downwards movement instead of going straight down when trying to land. This could be due to a lack of experience with VR or it could be because they may have been uncomfortable with going straight down. Another participant said they were missing a crosshair for this locomotion method, which could mean that there were a lack of feedfoward information. A lack of information could make it harder to predict movement which could result in an increase of VR sickness, if something very unexpected were to happen to the participant.

The last of the locomotion methods, Automove, had fewer observations that the rest of the locomotion methods. This could be due to the participants being told that they did not have control of movement or speed. Which may explain why most would stand completely still and look straight ahead or straight towards to the target while using this method. For the participants who started with the Automove methods were observed to look around as they descended while participants who had tried one or more of the other locomotion methods would look towards the target. Participant 8 was observed to wobble doing the Automove transition, this could have been because it was the first of the four locomotion method this participant would try. Meaning they may not know where they were going or where they would land. Participant 19 said after having tried Automove *"it was easy to steer but not easy to endure"*, which could indicate ease of use but this does not necessarily indicate preference.

Scenes

A few noticeable observations were done in Mountain, River, and Wastewater scenes. For the Mountain scene a participant would move around the physical world in order to try to impact the movement in the VRE. This extra movement could end up increasing a feeling of VR sickness in the participant, though there are little to no signs for this as their SSQ score in this scene is the second lowest. This little to no increase in VR sickness could be due to the participant moving around in the physical world and is prepared for any changes this may have caused. It may also help explain why the Mountain scene is one of the lowest scoring scenes of the six, as the participants would feel more connected or feel they had more agency compared to the other scenes. The River scene had two participants bend down under the water surface. This could have increased VR sickness though the participants that did that were around the middle of the River scene box plot, making it hard to discern any effects this may have had on them. It is worth noting that one of the participant stood up too quickly and almost lost their balance, that participant has the highest score for VR sickness of the two. A few participants comment that some of the turns were sharper than they would prefer and they felt like they were being dragged in a completely different direction out of the blue. This was also observed for the Wastewater scene, where the participant would move through various pipes and tanks. The unpredictable movement in the big tanks were commented on in the same manner as the River scene, but the pipes were not. This could be due to the openness of the tanks which would give not indication of movement whereas the participants could predict how they would move from the shape and bends of the pipes. Two participants would say *"Oh god"* on two different occasions in the scene. The first were going through multiple dirt planes and the second entered the first of the big round tanks. This could be signs of discomfort which could lead to an increase of VR sickness or maybe that the participant would stop using the VRE in the real world. The wastewater scene had two points at which the frame rate dropped below the wanted amount, thus the Vive returned to the idle screen with a message that the application was not responding, after roughly a second the experience returned. This rapid switching of screens and lose of immersion could increase the VR sickness or cause frustrations and/or confusion for the participants. These negative effects could have impacted the SSQ for the given scene negatively. For reference of this issue see Figure 4.50 and 4.51.

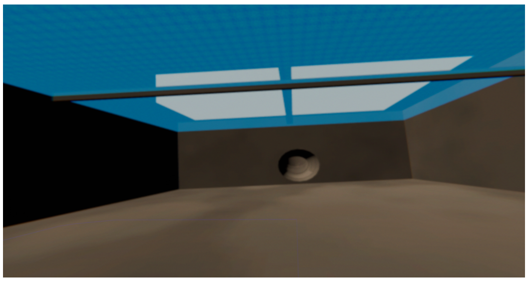

Figure 4.50: The view of a user in VR before the dropped framerate.

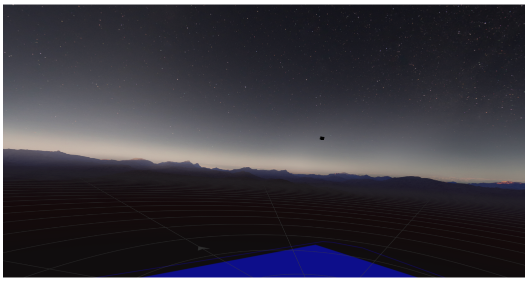

Figure 4.51: While the framerate is below the 45 frames per second the system switches to this idle screen.

Observations

When putting on the HMD three out of the 24 participants adjusted the distance between the eyes focus on the HMD. This could potentially increase the VR sickness when the participant did have a miss match regarding the image focus of the HMD or it could have increased the blurriness score in the SSQ question 11. *Blurred vision*.

Other general observations showed that only around two participants tried exploring the whole VRE instead of moving straight towards to target. Some participants commented that they thought the facilitators were keeping time of how long it would take them to move to the target. This was never stated in the description of the test or intentionally conveyed. It may explain why there were not more participants exploring the VRE using one or more of the locomotion methods. The thought of being timed may also have coursed stress for the participant which could lead to discomfort, stress, or sweating. Which may have effected different scores in both SSQ and CSUQ. Having most of the participants fly straight towards the target would help in getting a more homogeneous group, though it could also mean that the results could be less likely to show what participants prefer in a real world scenario.

The tendency of standing completely still was also observed in other locomotion methods and scenes were the participant were told that they would have no control of speed or movement, even though they could still move around physically and explore a small portion of the VRE that way. Some participants, though fewer than half, were observed to do at least one 180^{o} degree look around in the different scenes though still not moving much more than what was required to turn around. Not moving around the VRE could lead to a feeling of being dragged around and a lack of agency or connection to the VRE from the participants. This disconnection may appear in negative scores in the SSQ.

This test, Final, was conducted with the participant trying each locomotion method once and then trying one of the six scenes at the end as seen in appendix G. This could have been conducted in various other ways, which may have yielded other results as it could put the VRE in a different context. Another way the tests could have been conducted would be to have each participant try only one locomotion method and afterwards having them roam around the VRE as they please or potentially give them a simple task. As the VRE was build to be able to connect various scenes into a single or branching route, a task could be get to the ocean from the City scene. This test could put the VRE into a more realistic scenario where the participant would use it as they would in a real world scenario while not thinking to much about the individual locomotion methods. Some of the problems with the test form arises as it becomes harder to control various factors to create a homogeneous participant group. This test form gives a few questions which would needed to be addressed before use, some of these are: what should be done with participants who chose a wrong route, when should they be subjected by the various questionnaires, how often can their immersion be broken before it impacts the test. This test form would put more focus on the various scenes, in stead of the locomotion methods, as it is those which would be explored by the participant for the most of the VRE. Another testing method could be to test the scenes separate from the locomotion methods. This may remove any feeling VR sickness a participant may have gotten from the locomotion methods carrying over to the scenes and the SSQ about the scenes would not be affected by the noise. This was not done as it would take away some of the real world scenario the chosen test form had, where the participant would move straight to a scene after having landed or collided with various landmarks in the VRE.

The way the VRE was tested was in an effort to combine control of various factors with how the participants may use it in the real world. In an effort to subject each participant to the same minimum amount of VR sickness a specific target, as mentioned in Section 4.5.1, Test 5: Experiment, was chosen for them to hit. Not having the target could mean some participants would go to the Farm while others may just go straight down to the City. Leading to very different amount of experience with the VRE. It could also leading to a lose of focus whilst exploring and the participant forgetting what they were suppose to do. A timer could have been used in an effort to subject each participant to the same amount of VR sickness, though it would create problems for those locomotion methods with less controller for the participant. This could be the Teleport method, where the participants does not control when they land only where. Would they then be set back to the starting position or would they continue from where they are? This question also stands for the other locomotion methods, what should happen when they hit a landmark. It also leaves the question of should the participants be allowed to hit multiple landmarks or just one. Another problem with this test method could be that a 24 participant test is required for testing each combination sequence of the locomotion methods, which could lead to a problem of getting enough participants, as anything less than the 24 participants would lead to an unbalance for the locomotion methods. Looking at Figure 4.29 the scores seems to be in the lower end when compared to the box plots from the previous tests, Figure 4.6, 4.8, and 4.19. This could be due to participants only trying each locomotion method once instead of the three times which was seen in the other test methods. This could explain the lower scores but could also mean if the participant got to test each locomotion method more the score could either stay the same, get more equal to what was seen in the previous tests, or maybe exceed the scores of the previous tests. From Test 2 to 4 it would have seem that Teleport could have been the most preferred locomotion method, as it displayed some of the lowest scores in the SSQ. This could be due to the different methods the locomotion methods were compared to in the previous test. Given a test where one method being significantly worse than the other, may make the first method seem even better. Putting the same method in comparison to a different kind of locomotion method not significantly worse, may show a more true result. This could be the what is seen here when comparing the results from the previous test with the results from the final test.

4.6 General Observations

Through early testing it was found that participants were struggling with the virtual movement speed when close to surfaces of the world and objects. Since the speed was held constant, the participants mentioned that it was too fast when close to the surface and objects. Also participants were sometimes scared by the impact when colliding with the virtual worlds surface. Those two issues resulted in a script, that would measure the distance to objects around the participant in six different directions. If anything were closer than a set threshold the speed would slow down depend on how much over the threshold the object was to the participant. Meaning the closer to an object or the ground a participant were the slower they would go. This could cause rapid deceleration if the virtual world would change dramatically, resulting in start stop movement in a worst case scenario. A visualization of this problem can be seen in Figure 4.52.

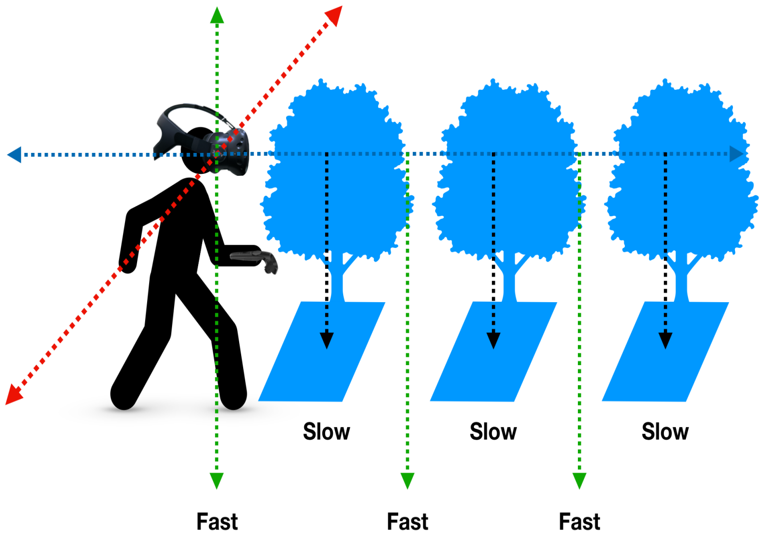

Figure 4.52: The red green and blue lines indicate the axis for which distance was measured. The black dotted lines visualize the circumstances that the threshold for distance was surpassed thereby speed got reduced.

In all tests a pre-SSQ was not given, this could mean that each participant that tried one version/method could have felt unwell before starting which may have led to biased results. Young et al, in the article "Demand Characteristics in Assessing Motion Sickness in a Virtual Environment: Or Does Taking a Motion Sickness Questionnaire Make You Sick?" found that in a between-group test taking a pre-SSQ increased the nausea and other symptoms the participants felt after trying VR, but not giving the participants a pre-SSQ means that a baseline for each participant cannot be established. To obtain a baseline Young et al. suggest basing it on a part of the population for the project and not each individual. Other than that it is suggested to figure out if all participants are healthy and well. Those who had an headache or otherwise showed signs of the symptoms from the SSQ would not try the tests. To get this information, a questionnaire could be used, observation, or asking the participants could also be suitable. As the SSQ asks about eye strain it could be argued that participants with glasses should not have been used as participants as they tend to remove their glasses when putting on the HMD. This may reduce their eye sight and force them to strain their eyes more. But it was deemed an acceptable noise factor, as it would make the population used for testing more true to what the product could meet in a real case scenario.

While testing the different versions of the chosen locomotion methods, Test 2 to 4, all participants were given the same three targets to hit each time. This was done in an attempt to have the participants move around for the same minimal amount of time and to get more homogeneous data sets to compare with each other. Giving these three target may have hindered the participants in exploring the locomotion methods as all target could be reached by going in a straight line from the starting position. Having a target to fly towards could also have caused the participant to focus on only hitting the given target which could have affected the results as they may prefer one form of locomotion methods for hitting a target and another for exploring the world.

Chapter 5

Conclusion

The problem statement of this project was to *Find a suitable navigation method for a multi-path Virtual Reality experience based on the hydrologic cycle* with the requirements of *Minimize VR sickness*, *Accurately track VR input*, and enable *Multiple paths*. Regarding the requirements it can be concluded that multiple paths were tested for transitioning points with the help of landmarks, also multiple scenes were tested, for which the outcome would require a more specific test though, due to the fact that the amount of participants which were used for test does not provide a solid database. In addition the general tendency towards freedom in all of the scenes was noted and might be key in a multi-pathing experience. Though the Automove methods are simple and effective, they are in general not the best at scoring low VR sickness results. Regarding the freedom thereby introduced to the user, it is noted that the speeds have to be adaptive and be decreasing the closer users come to objects. This could be due to speed being recognized relative, meaning that over high distances the movement seems slow while at close distance it seems to fast.

The initial and final test was assumed that the *Teleport Method* would generate the best outcome, since it initially performed best. There it was only tested against a highly similar method meaning that the context tested in might have an influence on the outcome as mentioned in Section 4.5.3, Test 5: Discussion. The generally most accepted and least VR sickness generating method for movement was noted to be the *Controller Method*.

For the level of immersion functionalities as special effects or sound might on one hand improve on the identification of being a water drop, on the other hand it can also lead to increasing VR sickness effects. This was noted from the Test 1: Experiment, Section 4.1.1, which showed that the average scored SSQ values nearly increased with 50%. For an implementation of camera effects the VRE would benefit from testing the combination of immersion effects.

From the data set of the Final test there could not be drawn any correlation between the *Scared of Heights*, the *Tendency towards Motion sickness*, and *Previous VR* *experience* for the outcome of the SSQ's. The highest outcome of the SSQ's was a person with the highest scores of *Medium* in both scared of heights and motion sickness questions. However, since other participants sometimes had one of the two set to *Medium* this persons results were seen as an outlier. Also the data set was not evenly distributed amongst gender, thereby this could not be used to generate any correlations between the SSQ's scores, test methods or scenes. From the SSQ's average values it can be seen that Trackpad and Controller perform better than the others. This complies with some individual responses that mentioned that the more freedom they had the less motion sick they get.

The results from the final CSUQ could be seen in Figure 4.30 to 4.38, where each locomotion method is compared with the others in each of the nine questions in the questionnaire. For the last test a significance value of 0,05 was set and a null hypothesis saying "The two locomotion methods are the same" is accepted if the alternative hypothesis is rejected. Starting with Teleport-Controller there is significantly different in five out of the nine questions: 1, 2, 4, 6, and 9. For Teleport-Trackpad it was only one question were a significant difference was found in question: 4. Lastly comparing Teleport-Automove there were a significance different two times with question: 2 and 9. Though question 5 was very close with a p-value of 0,05309, which could be seen as a significant different depended a decimal value. Next looking at the Controller-Trackpad there were calculated significant differences in five of nine questions: 1, 2, 6, 7, and 9. For the Controller-Automove comparisons the results were seven out of nine questions: 2, 3, 4, 5, 6, 8, and 9. Lastly comparing Trackpad-Automove another five out of nine questions were seen to be significantly different: 2, 3, 4, 5, and 9. All the values could be seen in Figure 4.40 to 4.48. From comparison of locomotion methods it would appear that Controller-Automove shows the most amount of difference, followed by Teleport-Controller, Controller-Trackpad, and lastly Trackpad-Automove all having five out of nine questions with significant difference. From the different pie chart, Figure 4.30 to 4.38, it is concluded the Controller is the overall best choice, as it is the highest scoring locomotion method in all but four questions where Automove is deemed the most preferred. This choice is even further backup by question 9. *Which method was overall the most satisfying?* where the Controller has the biggest score of both that question, and in all the rest, with 54,2% of all votes.

The two outliers regarding the Controller locomotion method were participant 23 and 24, who in general had above average scores, with at least double the general averages response score. Participant 23 replied for both "scared of heights" and "tendency towards motion sickness" with *Medium*. Participant 23 was moreover an outlier in Trackpad, Controller, and Automove, while being the top whiskers value for Teleport.

All in all it is concluded that the Controller locomotion method was the best suited method for the VRE out of all four methods tested. It showed some of the lowest scoring when compared to the other locomotion method in the SSQ and was seen as one of the best preforming in CSUQ. The SSQs indicate that the Controller is the method which performed best in regards to VR sickness. Not only has the controller the lowest median value, it also has the most compact box, declaring the least amount of variance among the participants. The Controller method had the general lowest value of zero still in its box, displaying that it was among the 75th percentile of responses. It is believed this is due to the amount of control available to the participants when using this locomotion methods, with both direction and when to move. The Controller method may not give the participants an ability to control the speed it may be slow enough for most participants. This speed combined with the dynamic speed system implemented seems to be enough for it to be comfortable. It may be that the Trackpad method gives too much control which would lead to more for the participant to lean and combining control of speed with the dynamically changing speed may be the reason for more VR sickness.

It should be noted that the score seen from Test 3: Controller, where two similar Controller locomotion methods was compared, was not as good as seen in the Final test. Resulting in the conclusion the preference of the locomotion methods are subject to what they are being compared to.

Chapter 6

Limitations & Future Work

This Chapter evaluates what this report could have improved on also it displays how the work of this report could be extended in future projects, to benefit the primary goal of creating a virtual reality user experience, that would inform about the hydrologic cycle.

6.1 Limitations

For this project the amounts of tests were necessary to generate a data set which could indicate preferences of the participants. It has to be noted that the participant group this project focused on was between the age of 20 to 30 while there were two outliers at the age of 32 and 35. In the case where the VRE should be used for educational purposes and it is supposed to work with a different focus group, the major tests would then require retesting, but the results of this study could be used as baseline.

The outcome of the CSUQ from the final test indicate that the method chosen to ask the questions was not generating the most suitable results for the comparison. Although it was necessary due to the length of the test in order to find participants. For a higher precision on this test, a 7 step likert scale regarding every individual method. This would generate a data set which then would not get swallowed by other categories, as the *AutoMove* does for four of the simplicity questions. The tests could also have been made with an initial baseline of a pre-SSQ, as mentioned in Section 4.6, General Observations. Which then could potentially reduce noise from the data. The issue with this pre-SSQ would be that it might amplify the outcome of the tests, which has to be considered if it would be compared.

Regarding the program used for testing there were issues that might have increased the effect of VR sickness obtained by a test. For instance the fadeout which is used for the main scene that has the issue of fading in again if the participant has moved past the ground. This might have increased the VR sickness score, this noise was accepted for this project, since none of the participants complained about the fadeout problem. Continuing with the application problems the frame rate issue in the *Water Treatment* scene could have increased the VR sickness dramatically. This potential increase could be caused by the SteamVR idle screen that had no transition except a frozen image. Due to this issue and the low testing count of four participants per scene, the data did not provide any reliable information. Lastly all the followup scenes were using a auto moving method that had points distributed along the scene, in order verify that the participants had the chance to explore everything along a predetermined route. This generates an issue, as the questionnaires and the individual comments suggest, that the participants did show a tendency to dislike it when moved automatically. Even though the participants still could alter the placement on the route by moving inside the tracked area, the route would still continue towards those points. In addition the curves when using those movement methods it was commented that the angles were to sharp.

6.2 Future work

For future development in the scope of this project, a higher level of immersion would potentially be helpful to gather more relevant data for the use case of the educational VRE for Grundfos. For an increase in immersion, sound would be the next major step to test, in combination with defining a common ground for the camera perspective that has to reassemble water. The sound effects may have to vary, to such an extend that it is delivered dynamically to the user depending on the action. In cases where the water drop falls the wind or motion sound would have to increase, while in cases where the water surface changes, the sound would have to be responsive and might change the pitch value as well, which might provide a underwater effect. Sounds could also be used for an audio guide which might not only explain information regarding the hydrologic cycle but also be a warning for sudden movement preparing the user on drastic changes. Regarding the camera effects, speed effects for falling could potentially provide an effect on VR sickness as well as it might mislead the user to think that they move faster or slower than they actually are. Also methods such as shaking cameras which are used in video games or movies when something is falling should either be avoided or be tested to see how huge an impact this would have on VR sickness, compared to the level of immersion gained.

Depending on the different scenarios of the hydrologic cycle a variety of new scenarios could be created and combined with multi-pathing of the VRE in order to provide a better educational purpose. Also a new movement method in other scenes would be required since the static behaviour in other scenes were in general disliked. In addition to the point to point movement indication of where to go with a feed forward system could improve the the SSQ score. In relation to the water treatment plant air bobbles inside the water indicating the flow stream could lead to a higher perception of how the user would move.

Regarding the testing, the data outcome would highly benefit from a larger data set, and it might also be useful to use a different method for testing the individual points of focus. The methods as suggested in Section 4.5.3, Test 5: Discussion, might provide more insight as well. In addition the water on screen effects might be helpful with the level of immersion. Though it would need either adaption or combination with other immersion benefiting methods to be useful for the VRE and avoid an increase of VR sickness. Also the amount of participants would generally improve the credibility of each test, meaning that outliers could be found more precisely.

For the CSUQ it could be beneficial to provide the participants with a seven step "strongly agree" to "strongly disagree" likert scale for each of the locomotion methods to get higher precision on the outcome of the study. Moreover the amount of female participants amongst the final test was too low and would need to be tested with a evenly distributed set.

Other movement methods which might use secondary devices, such as treadmills which where mentioned in Chapter 2, Background theory, could potentially also benefit the case of representing water. This could for instance reduce the amount of VR sickness which the users would be exposed to by the locomotion methods.

Lastly if further development added other ways of interacting with the VRE, other than moving around. It could be the results found here are less reliable, as they were tested as the only interaction method and are created with that in mind. Leading to if an user would have to grab on to something, they would it only be able to use one hand or switch between locomotion and interaction. Thereby depending on all the actions the user would have to do, the VRE might need different or slightly different locomotion methods and interaction controls which seem intuitive.

Bibliography

[1] *About Grundfos*. https://www.grundfos.com/about-us.html. Accessed: 06/02/2019.

[2] Assila Ahlem, Káthia M. d. Oliveira, and Houcine Ezzedine. "Standardized Usability Questionnaires: Features and Quality Focus". In: *Electronic Journal of Computer Science and Information Technology : eJCIST* (2016).

[3] A. M. Gavgani et al. "Effects of visual flow direction on signs and symptoms of cybersickness". In: (2017).

[4] Qi Sun et al. "Towards Virtual Reality Infinite Walking: Dynamic Saccadic Redirection". In: (2018).

[5] Robert S. Allison et al. "Tolerance of Temporal Delay in Virtual Environments". In: *Proceedings IEEE Virtual Reality 2001* (2001). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=913793.

[6] Audodesk. *About us*. https://www.autodesk.com/company#. Accessed: 27/02/2019.

[7] Audodesk. *Oversig*. https://www.autodesk.dk/products/maya/overview. Accessed: 28/02/2019.

[8] Rafael Ballagas and Steffen P. Walz. "REXplorer: Using player-centered iterative design techniques for pervasive game development". In: (). Accessed: 12/02/2019.

[9] Shaun Bangay and Louise Preston. "An Investigtion Into Factors Influencing Immersion in Interactive Virtual Reality Environments". In: (1998). Accessed: 18/02/2019.

[10] *Base Station*. https://www.vive.com/eu/accessory/base-station/. Accessed: 27/02/2019. HTC Corporation.

[11] Chris Bateman. *Imiginary Games*. Zero Books, 2011, pp. 31–37.

[12] *BodyNav XR Developer Kit*. https://monkey.com/xdk/. Accessed: 13/02/2019. 2018.

[13] Costas Boletsis. "The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology". In: (2017).

[14] Jay Collins et al. "EDTree: Emotional Dialogue Trees for Game Based Training". In: (). Accessed: 14/02/2019.

[15] English Oxford Living Dictionaries. *Immersion*. https://en.oxforddictionaries.com/definition/immersion. Accessed: 07/02/2019.

[16] English Oxford Living Dictionaries. *Transforming our world: the 2030 Agenda for Sustainable Development*. https://sustainabledevelopment.un.org/post2015/transformingourworld. Accessed: 07/02/2019.

[17] JAMES C.I. DOOGE. "THE HYDROLOGIC CYCLE AS A CLOSED SYSTEM". In: *International Association of Scientific Hydrology. Bulletin* 13.1 (1968), pp. 58–68. DOI: 10.1080/02626666809493568. URL: https://doi.org/10.1080/02626666809493568.

[18] Ernst Kruijff Doug A. Bowman and Ivan Poupyrev Joseph J. LaViola. *3D User Interfaces: Theory and Practice*. 2. ed. Addison Wesley, 2017, pp. 324–371.

[19] Edulearn. *What is Autodesk Maya? How is Maya used?* https://www.edulearn.com/article/what_is_autodesk_maya.html. Accessed: 28/02/2019.

[20] *Explore Unity*. https://unity3d.com/unity. Accessed: 27/02/2019. Unity Technologies.

[21] Tom Forsyth. *Connect: Developing VR Experiences with the Oculus Rift*. https://www.youtube.com/watch?v=addUnJpjjv4&feature=youtu.be&t=40m5s. time stamp: 40:06-41:41, Accessed: 06/02/2019.

[22] Tracy Fullerton. *GAME DESIGN WORKSHOP, A Playcentric Approach to Creating Innovative Games*. 3. ed. A K Peters/CRC Press, 2014.

[23] *GameObject*. https://docs.unity3d.com/ScriptReference/GameObject.html. Accessed: 04/03/2019. Unity Technologies.

[24] Reginald G. Golledge. *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. The Johns Hopkins University Press, 1999, pp. 16–19.

[25] *Google Earth VR*. https://store.steampowered.com/app/348250/Google_Earth_VR/. Accessed: 01/04/2019. Google.

[26] *GRAND THEFT AUTO: VICE CITY*. https://www.rockstargames.com/games/info/vicecity. Accessed: 29/05/2019. Rockstar Games.

[27] Grundfos. *Optimised water solutions*. http://www.watercycle.grundfos.com/?fbclid=IwAR0qpQOHZiW9MTsyTq81c0wZb5dJKzGlM95gJs2xwVr3FskbMZ6_Dz11QIY#/05_waste-water-treatment/-/standard-layer. Accessed: 14/04/2019.

[28] *Guidelines for VR Performance Optimization*. https://developer.oculus.com/documentation/pcsdk/latest/concepts/dg-performance-guidelines/. Accessed: 13/02/2019. Oculus, Facebook Technologies, LLC.

[29] Jason Jerald. *The VR book : Human-Centered Design for Virtual Reality*. Morgan & Claypool, 2016.

[30] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *International Journal of Aviation Psychology* (1993).

[31] Eugenia M. Kolasinski. "Simulator Sickness in Virtual environments". In: (1995). https://apps.dtic.mil/dtic/tr/fulltext/u2/a295861.pdf.

[32] *Leading the XR Revolution*. https://unity3d.com/unity/features/multiplatform/vr-ar. Accessed: 27/02/2019. Unity Technologies.

[33] Dong-Yong Lee, Yong-Hun Cho, and In-Kwon Lee. "Being Them: Presence of Using Non-human Avatars in Immersive Virtual Environment". In: *VRST '18 Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology* (2018). Accessed: 25/02/2019.

[34] Juyong Lee, Sang Chul Ahn, and Jae-In Hwang. "A walking-in-Place Method for Virtual Reality Using Position and Orientation Tracking". In: (2018). Accessed: 26/02/2019.

[35] Michael van Lent et al. "Increasing Replayability with Deliberative and Reactive Planning". In: (2005). http://www.aaai.org/Papers/AIIDE/2005/AIIDE05-023.pdf.

[36] Antonella Maselli and Mel Slater. "The building blocks of the full body ownership illusion". In: (2013). Accessed: 18/02/2019.

[37] Daniel Medeiros et al. "Effects of Speed and Transitions on Target-based TRavel Techniques". In: *VRST '16 Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (2016). Accessed: 26/02/2019, pp. 327–328.

[38] *MONKEYmedia Addresses Motion Sickness in Virtual Reality with Release of Developer Toolkit*. https://www.businesswire.com/news/home/20180925005425/en/. Accessed: 25/04/2019. 2018.

[39] *Movement in VR*. https://unity3d.com/learn/tutorials/topics/virtual-reality/movement-vr. Accessed: 09/04/2019. Unity Technologies.

[40] United Nations. *Sustainable Development Goals*. https://sustainabledevelopment.un.org/sdgs. Accessed: 05/02/2019.

[41] United Nations. *Sustainable Development Goals 6*. https://www.un.org/sustainabledevelopment/water-and-sanitation/. Accessed: 05/02/2019.

[42] *Oculus Rift*. https://www.oculus.com/rift/#oui-csl-rift-games=star-trek. Accessed: 12/02/2019.

[43] Petter Ögren et al. "Design and implementation of a new teleoperation control mode for differential drive UGVs". In: *Autonomous Robots* (2014). Accessed: 28/02/2019.

[44] Donald E. Parker. *Reviewed Work: Motion Sickness by J. T. Reason, J. J. Brand*. https://www.jstor.org/stable/1421659?sid=primo&origin=crossref&seq=1#metadata_info_tab_contents. 1977.

[45] *PlayStation VR*. https://www.playstation.com/en-ae/explore/playstation-vr/. Accessed: 12/02/2019.

[46] Bernhard E. Riecke et al. "Do We Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice". In: (2010).

[47] Jay Samit. *A Possible Cure for Virtual Reality Motion Sickness*. http://fortune.com/2018/02/06/virtual-reality-motion-sickness/. Accessed: 12/02/2019. 2018.

[48] *Steam Hardware & Software Survey: January 2019*. https://store.steampowered.com/hwsurvey. Accessed: 27/02/2019. Valve Corporation.

[49] *SteamVR*. https://steamcommunity.com/steamvr. Accessed: 27/02/2019. Valve Corporation, 2015.

[50] *SteamVR Plugin*. https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647. Accessed: 27/02/2019. Valve Corporation, 2015.

[51] Techopedia. *Non-Player Character (NPC)*. https://www.techopedia.com/definition/1920/non-player-character-npc. Accessed: 14/02/2019.

[52] Daniel Terdiman. *And the Oscar for best visual effect goes to... Autodesk's Maya*. https://venturebeat.com/2015/01/15/hollywood-fx-pros-i-want-to-be-an-oscars-maya-winner/. Accessed: 28/02/2019.

[53] *The Lab*. https://store.steampowered.com/app/450390/The_Lab/. Accessed: 01/04/2019. Valve.

[54] Mathew Tomberlin, Liudmila Tahai, and Krzysztof Pietroszek. "Gauntlet: Travel Technique for Immersive Environments using Non-dominant Hand". In: (2017). Accessed: 26/02/2019, pp. 299–300.

[55] tv tropes. *Non-player Character aka: NPC*. https://tvtropes.org/pmwiki/pmwiki.php/Main/NonPlayerCharacter?from=Main.NPC. Accessed: 14/02/2019.

[56] Unity. *Movement in VR*. https://unity3d.com/learn/tutorials/topics/virtual-reality/movement-vr. Accessed: 06/02/2019.

[57] *Unity*. https://unity3d.com/. Accessed: 27/02/2019. Unity Technologies.

[58] *Vive*. https://www.vive.com/eu/. Accessed: 12/02/2019.

[59] *VIVE VR SYSTEM*. https://www.vive.com/us/product/vive-virtual-reality-system/. Accessed: 27/02/2019. HTC Corporation.

[60] *VRTK - Virtual Reality Toolkit - [VRToolkit]*. https://assetstore.unity.com/packages/tools/integration/vrtk-virtual-reality-toolkit-vr-toolkit-64131. Accessed: 26/04/2019. Extend Reality Ltd.

[61] Brooklyn Waters. "Physics and Frame Rate: Beating motion sickness in VR". In: (2015). http://mtechgames.com/downloads/PhysicsandFramerateBeating motionsicknessinVR.pdf.

Appendix A

Participant Info and Simulation Sickness Questionnaire

VR Teleportaion locomotion test

Thank you very much for your help in testing this Virtual Reality (VR), by answering this questionnaire you agree to letting the data collected from the test be used in group 1041 VGIS master's thesis. All data collected will only be used to for the project and will not be used for anything else.

I agree to let group 1041 VGIS10 use the data colledted here for their master projects:
O: Yes
O: No

Sex:
O: Famele
O: Male

Age:

____

Previous experience with VR:
O: None
O: A little
O: Medium
O: A lot

Simulation Sickness

Please pick the answer for each question that you feel is most accurate for what you felt while in the VR simulation. If you have any questions please ask one of the facilitators.

1. General discomfort
O: None
O: Slight
O: Moderate
O: Severe

2. Fatigue
O: None
O: Slight
O: Moderate
O: Severe

3. Headache
O: None
O: Slight
O: Moderate
O: Severe

4. Eye strain
O: None
O: Slight
O: Moderate
O: Severe

5. Difficulty focusing
O: None
O: Slight
O: Moderate
O: Severe

6. Salivation increasing
O: None
O: Slight
O: Moderate
O: Severe

7. Sweating
O: None
O: Slight
O: Moderate
O: Severe

8. Nausea
O: None
O: Slight
O: Moderate
O: Severe

9. Difficulty concentrating
O: None
O: Slight
O: Moderate
O: Severe

10. "Fullness of head"
O: None
O: Slight
O: Moderate
O: Severe

11. Blurred vision
O: None
O: Slight
O: Moderate
O: Severe

12. Dizziness with eyes open
O: None
O: Slight
O: Moderate
O: Severe

13. Dizziness with eyes closed
O: None
O: Slight
O: Moderate
O: Severe

14. Vertigo
The experience of loss of orientation with respect to vertical upright
O: None
O: Slight
O: Moderate
O: Severe

15. Stomach awareness
A feeling of discomfort which is just short of nausea
O: None
O: Slight
O: Moderate
O: Severe

16. Burping
O: None
O: Slight
O: Moderate
O: Severe

Appendix B

Computer System Usability Questions

Comparing the two versions

Please pick the answer for each question that you feel is most accurate for what you felt while in the VR simulation.

1. Which method was easier to use?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

2. Which method was simpler to use?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

3. Which method made it more effective completing the tasks?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

4. Which method made it quicker to complete the tasks?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

5. Which method made it more efficient to complete the tasks?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

6. Which method was most comfortable to use?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

7. Which method was easier to learn to use?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

8. Which method behaved must like expected?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

9. Which method was overall most satisfying?
O: Very much Controller
O: Slightly Controller
O: Neutral
O: Slightly Display
O: Very much Display

Appendix C

Controller test Computer System Usability Questions Table

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
P 01	-2	-1	1	2	0	-2	-1	-1	-1
P 02	-1	-1	-2	0	0	-2	-2	-1	-1
P 03	1	-1	2	2	2	1	1	2	2
P 04	0	-1	-2	-1	-1	0	-2	-2	-1
P 05	1	1	1	2	2	1	1	2	1
P 06	0	0	0	0	1	2	-1	-1	1
P 07	-2	-1	0	0	0	-2	-1	-2	-2
P 08	-1	-1	-1	0	-1	-2	-1	0	-1
P 09	-2	1	2	-2	-2	1	1	0	1
P 10	-2	-1	0	-1	0	-2	-2	-1	-1
P 11	-2	-2	-2	-2	-2	-2	-2	-2	-2
P 12	1	1	1	1	1	2	2	2	2
P 13	-2	1	-1	1	0	-1	1	1	-1
P 14	-2	-2	-1	-2	-2	-1	-2	-2	-2
P 15	-2	-2	0	-1	-1	2	-2	-2	1
P 16	-1	-2	-1	-1	-1	-1	-2	-1	-1
P 17	1	-1	2	0	0	0	-1	1	0
P 18	-2	-1	-2	-2	-2	-2	-2	-1	-2
P 19	1	-1	-1	-1	-1	-2	0	0	-2
P 20	0	1	0	0	1	1	0	-1	1
P 21	2	0	0	-2	0	2	1	1	2

The raw date from the CSUQ questionnaire from test 3 Controller. The top grey row is the questions, Q1-Q9, and along the left hand column are the participants, P 01-P 21. Figure C.1 to C.9 shows a bar graph made from each question from table.

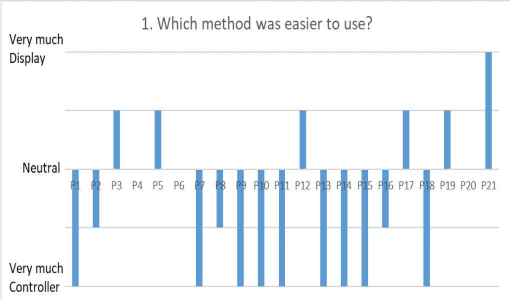

Figure C.1: Question 1. from the CSUQ

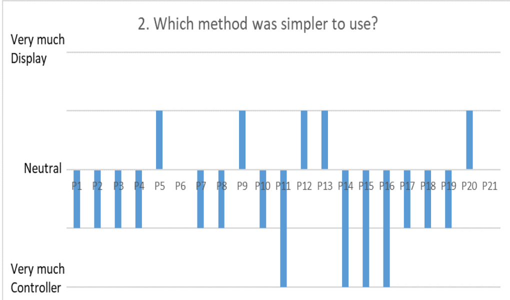

Figure C.2: Question 2. from the CSUQ

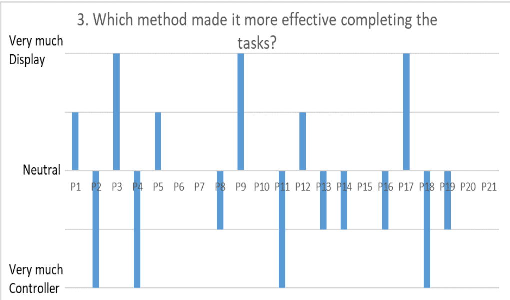

Figure C.3: Question 3. from the CSUQ

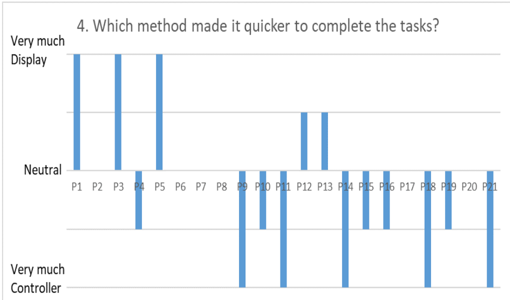

Figure C.4: Question 4. from the CSUQ

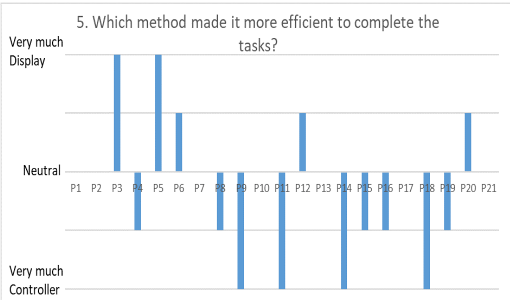

Figure C.5: Question 5. from the CSUQ

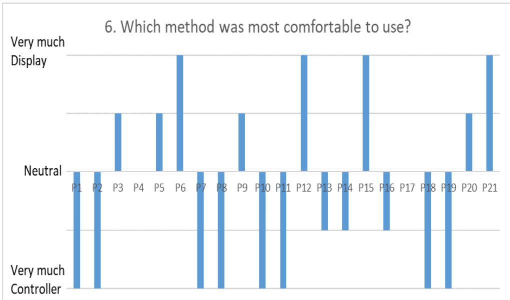

Figure C.6: Question 6. from the CSUQ

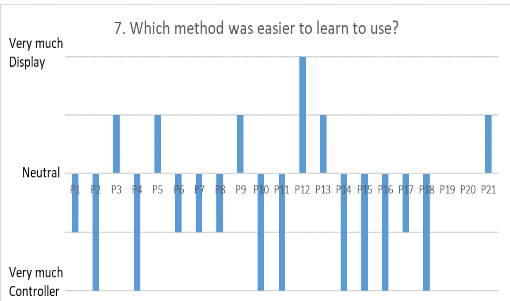

Figure C.7: Question 7. from the CSUQ

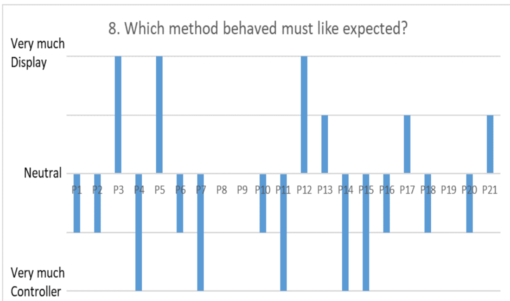

Figure C.8: Question 8. from the CSUQ

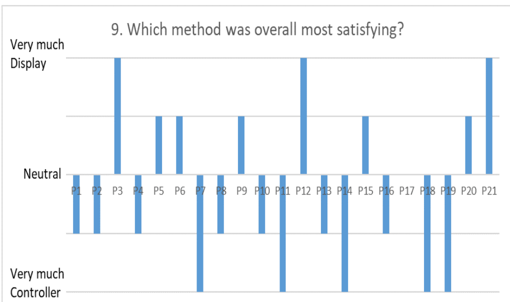

Figure C.9: Question 9. from the CSUQ

Appendix D

Trackpad test Computer System Usability Questions Table

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
P 01	1	-1	2	1	2	0	-1	1	0
P 02	-2	-1	-1	-1	-2	-2	0	-2	-2
P 03	1	0	-1	-2	1	1	2	1	1
P 04	-2	-1	-2	-2	-2	0	-2	-2	-1
P 05	1	-1	1	1	-1	1	0	2	1
P 06	-1	-2	-2	-2	-2	-2	-2	-2	-2
P 07	-1	-1	0	1	1	-1	0	0	-1
P 08	2	2	2	2	2	2	2	2	2
P 09	-1	-1	-1	-1	-1	-2	0	0	-1
P 10	-1	-1	-2	-1	-2	0	-1	-1	-2
P 11	1	1	1	2	1	1	0	0	1
P 12	-2	-2	-2	-2	-2	0	-2	2	2
P 13	-2	-2	-1	2	-2	-2	2	-1	-2
P 14	0	0	-2	-2	-2	-2	1	0	-2
P 15	0	1	1	1	1	1	1	1	0
P 16	-2	-1	-2	-2	-2	-2	0	-1	-2
P 17	-1	-1	0	0	0	-1	1	0	-1
P 18	-2	-2	-2	-2	-2	-2	0	1	-2
P 19	1	2	1	1	1	2	2	0	2
P 20	1	2	1	0	0	1	1	-1	1
P 21	1	1	2	2	2	1	1	1	1
P 22	1	1	2	1	1	2	1	2	1

The raw date from the CSUQ questionnaire from test 4 Traackpad. The top grey row is the questions, Q1-Q9, and along the left hand column are the participants, P 01-P 22. Figure D.1 to D.9 shows a bar graph made from each question from table.

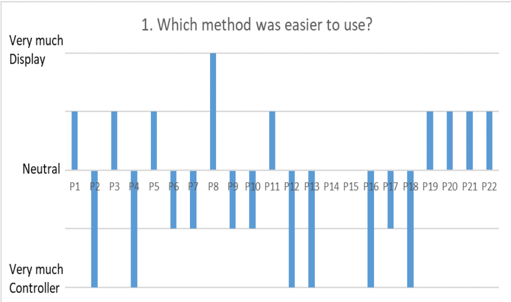

Figure D.1: Question 1. from the CSUQ

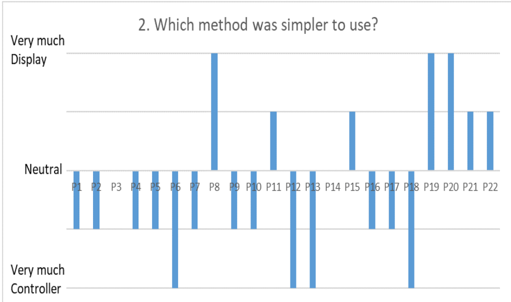

Figure D.2: Question 2. from the CSUQ

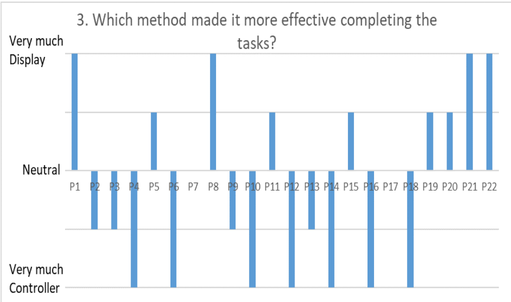

Figure D.3: Question 3. from the CSUQ

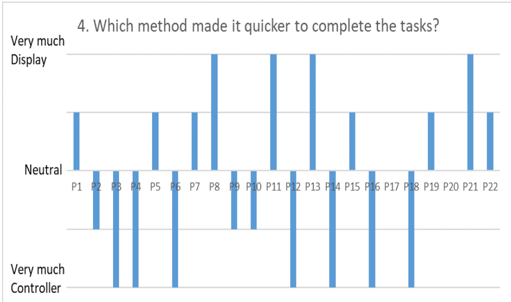

Figure D.4: Question 4. from the CSUQ

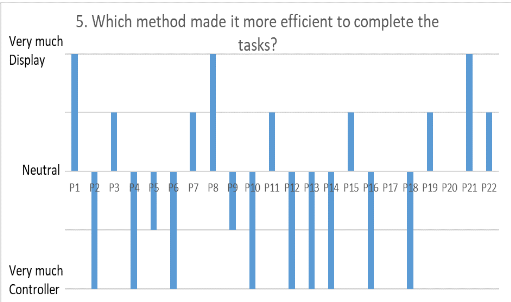

Figure D.5: Question 5. from the CSUQ

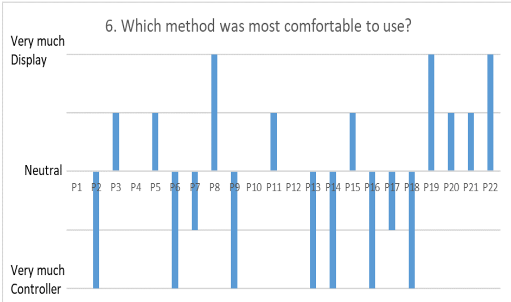

Figure D.6: Question 6. from the CSUQ

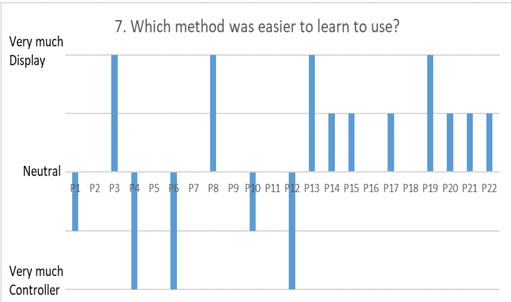

Figure D.7: Question 7. from the CSUQ

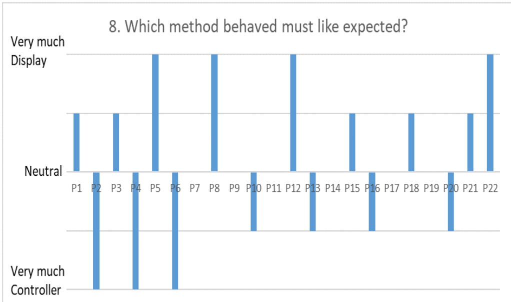

Figure D.8: Question 8. from the CSUQ

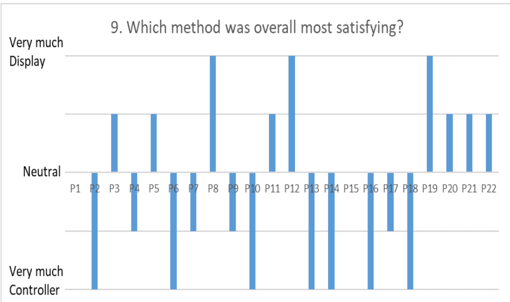

Figure D.9: Question 9. from the CSUQ

Appendix E

Final test Computer System Usability Questions

Comparing the two versions

Please pick the answer for each question that you feel is most accurate for what you felt while in the VR simulation.

1. Which method was easier to use?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

2. Which method was simpler to use?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

3. Which method made it more effective completing the tasks?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

4. Which method made it quicker to complete the tasks?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

5. Which method made it more efficient to complete the tasks?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

6. Which method was most comfortable to use?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove

O: Slightly Automove
O: Neutral

7. Which method was easier to learn to use?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

8. Which method behaved must like expected?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

9. Which method was overall most satisfying?
O: Very much Teleport
O: Slightly Teleport
O: Very much Controller
O: Slightly Controller
O: Very much Trackpad
O: Slightly Trackpad
O: Very much Automove
O: Slightly Automove
O: Neutral

Appendix F

Final test Computer System Usability Questions Table

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
P1	8	1	7	8	8	4	8	1	3
P2	8	2	3	7	3	8	3	4	4
P3	2	2	2	2	3	1	1	6	6
P4	1	2	1	7	7	4	7	8	6
P5	2	2	1	8	1	3	0	3	4
P6	5	1	5	5	5	1	6	5	5
P7	2	2	8	8	8	8	8	6	8
P8	1	1	8	7	1	7	5	4	8
P9	6	5	6	8	6	6	7	7	6
P10	6	2	2	1	1	6	5	6	6
P11	5	5	5	2	2	5	5	5	7
P12	5	7	2	2	1	6	2	2	6
P13	6	2	8	8	2	6	5	6	6
P14	6	2	2	2	2	5	2	0	6
P15	2	2	3	6	2	6	6	6	6
P16	5	2	6	7	7	5	5	0	3
P17	4	5	4	5	5	3	5	5	4
P18	8	6	8	8	8	6	8	8	6
P19	6	6	2	2	2	6	6	6	6
P20	4	7	2	2	1	4	7	4	4
P21	5	5	8	8	7	5	5	0	8
P22	6	1	1	1	1	6	1	0	6
P23	3	2	2	1	1	3	3	3	3
P24	2	5	2	2	2	6	2	8	6

The raw date from the CSUQ questionnaire from test 5 Final. The top grey row is the questions, Q1-Q9, and along the left hand column are the participants, P 01-P 24. Figure F.1 to F.9 shows Bar graphs made from each question from table.

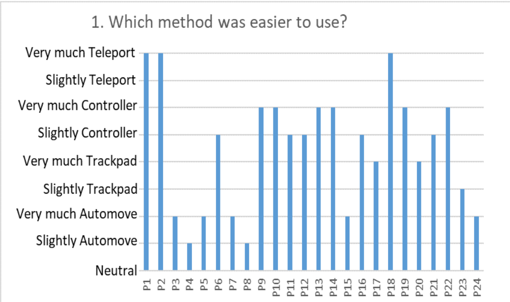

Figure F.1: Question 1. from the CSUQ

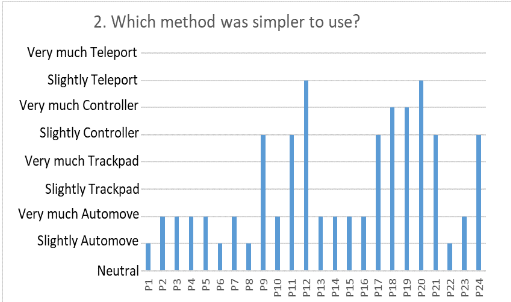

Figure F.2: Question 2. from the CSUQ

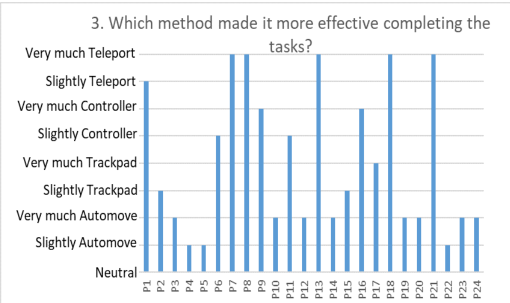

Figure F.3: Question 3. from the CSUQ

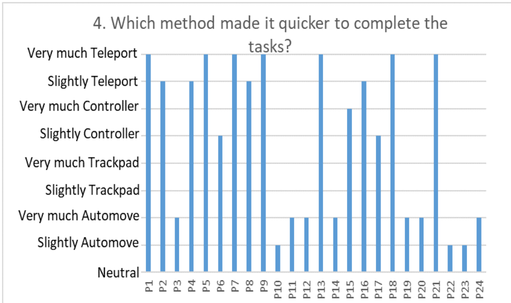

Figure F.4: Question 4. from the CSUQ

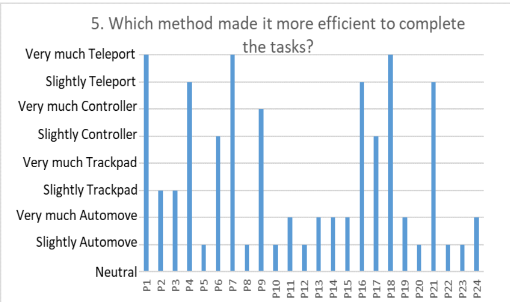

Figure F.5: Question 5. from the CSUQ

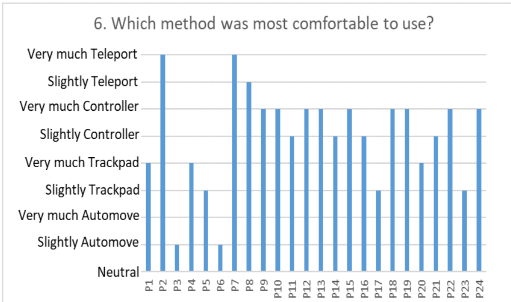

Figure F.6: Question 6. from the CSUQ

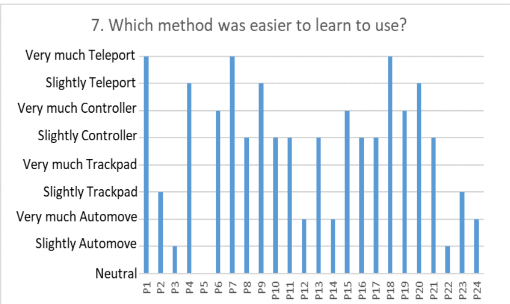

Figure F.7: Question 7. from the CSUQ

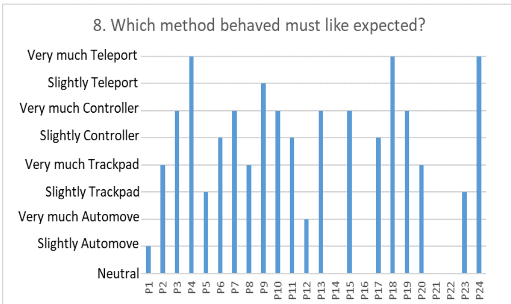

Figure F.8: Question 8. from the CSUQ

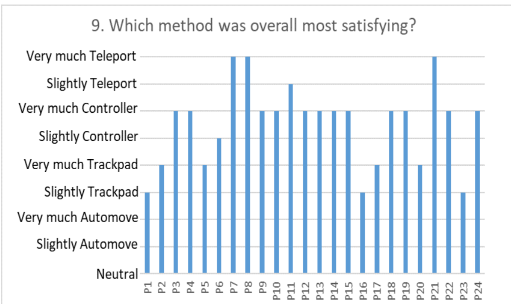

Figure F.9: Question 9. from the CSUQ

Figure F.10 to F.18 shows Histogram graphs made from each question from table.

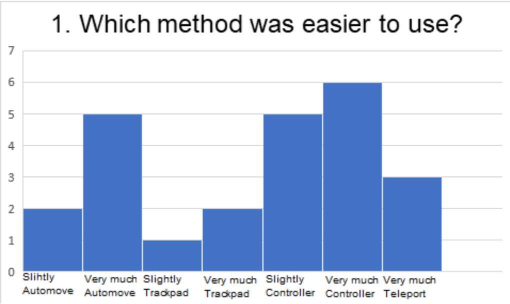

Figure F.10: Question 1. Histogram from the CSUQ

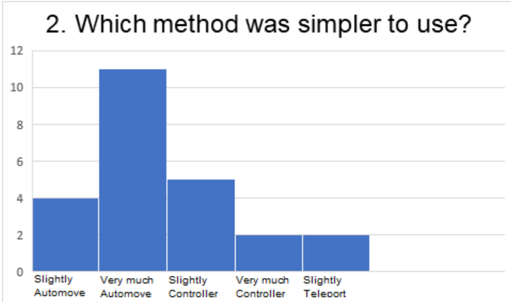

Figure F.11: Question 2. Histogram from the CSUQ

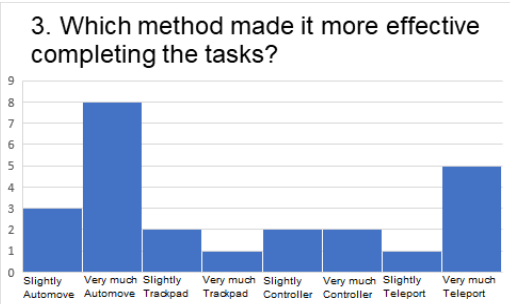

Figure F.12: Question 3. Histogram from the CSUQ

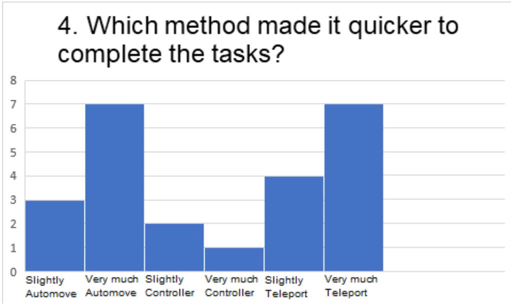

Figure F.13: Question 4. Histogram from the CSUQ

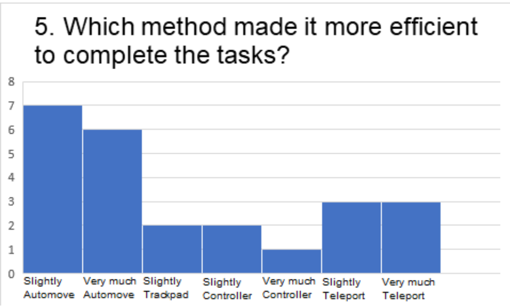

Figure F.14: Question 5. Histogram from the CSUQ

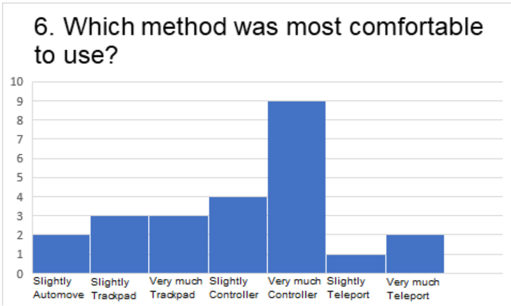

Figure F.15: Question 6. Histogram from the CSUQ

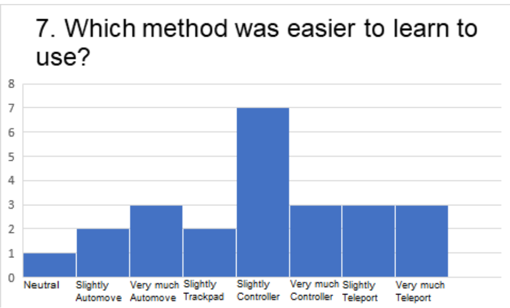

Figure F.16: Question 7. Histogram from the CSUQ

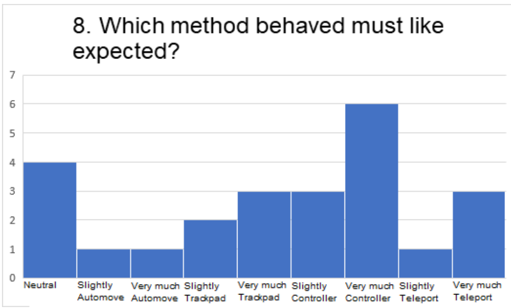

Figure F.17: Question 8. Histogram from the CSUQ

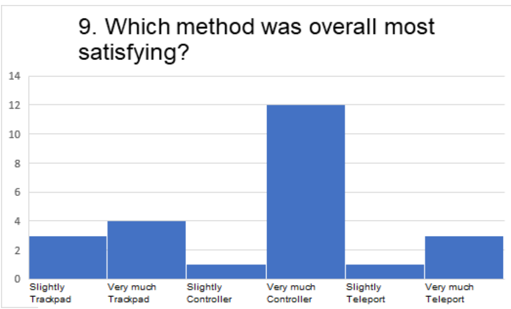

Figure F.18: Question 9. Histogram from the CSUQ

Appendix G

Testing Table for the final test

	Round 1	Round 2	Round 3	Round 4	Scene
Participant 1	Teleport	Trackpad	Controller	Auto	City
Participant 2	Teleport	Trackpad	Auto	Controller	Dessert
Participant 3	Teleport	Controller	Trackpad	Auto	Farm
Participant 4	Teleport	Controller	Auto	Trackpad	Mountain
Participant 5	Teleport	Auto	Controller	Trackpad	River
Participant 6	Teleport	Auto	Trackpad	Controller	Wastewater
Participant 7	Trackpad	Teleport	Controller	Auto	City
Participant 8	Trackpad	Teleport	Auto	Controller	Dessert
Participant 9	Trackpad	Controller	Teleport	Auto	Farm
Participant 10	Trackpad	Controller	Auto	Teleport	Mountain
Participant 11	Trackpad	Auto	Teleport	Controller	River
Participant 12	Trackpad	Auto	Controller	Teleport	Wastewater
Participant 13	Controller	Teleport	Trackpad	Auto	City
Participant 14	Controller	Teleport	Auto	Trackpad	Dessert
Participant 15	Controller	Trackpad	Teleport	Auto	Farm
Participant 16	Controller	Trackpad	Auto	Teleport	Mountain
Participant 17	Controller	Auto	Teleport	Trackpad	River
Participant 18	Controller	Auto	Trackpad	Teleport	Wastewater
Participant 19	Auto	Teleport	Trackpad	Controller	City
Participant 20	Auto	Teleport	Controller	Trackpad	Dessert
Participant 21	Auto	Trackpad	Teleport	Controller	Farm
Participant 22	Auto	Trackpad	Controller	Teleport	Mountain
Participant 23	Auto	Controller	Teleport	Trackpad	River
Participant 24	Auto	Controller	Trackpad	Teleport	Wastewater

Figure G.1: The scheme the participants were tested in to avoid influence of the locomotion methods on the following methods.

Appendix H

Resources Used for Scene Creation

The resources listed in this appendix are not created by the authors of this project. All credits for these resources should go to their creators, who can be found following the links.

Object Files:

- The Cactus
  https://poly.google.com/view/1_3Ur2-LAs8
- The Pine Tree
  https://poly.google.com/view/2Qo-fmVKuSG
- Apple tree
  https://poly.google.com/view/cpg6wP8JCyQ
- Tree
  https://poly.google.com/view/2acu5nrdDYl
- Stone1
  https://poly.google.com/view/dmRuyy1VXEv
- Stone2
  https://poly.google.com/view/3FmsLxIx8Lc
- Corn
  https://poly.google.com/view/dE73diXi82k
- Bridge 1
  https://poly.google.com/view/9oToSb_rBKY
- Bridge 2
  https://poly.google.com/view/1uQSi6qTp6o

- River vegetation 1
  https://poly.google.com/view/9uT74BMpRrl
- Tractor
  https://poly.google.com/view/5TGoA5N14c5

Textures

- Grass texture
  https://www.flickr.com/photos/wwarby/14866392570
- Dirt texture
  https://bit.ly/2WlhBrk
- Rock texture
  https://bit.ly/2VZ7Yz6
- Water texture
  https://bit.ly/30BE1E6
- Sand Texture
  https://commons.wikimedia.org/wiki/File:Sand.jpg
- Asphalt Texture
  https://upload.wikimedia.org/wikipedia/commons/thumb/5/5b/Asphalt_concrete.JPG/1599px-Asphalt_concrete.JPG
- Water Drop Texture
  http://pngimg.com/download/3317
- Toon Water Texture
  https://cdn.pixabay.com/photo/2015/11/02/18/32/water-1018808_1280.jpg

Shaders

- Heat distortion
  https://github.com/vux427/ForceFieldFX/blob/master/ForceFieldFX/Assets/Shader/ShieldFX.shader
- Sphere inner shader
  https://answers.unity.com/questions/176487/materialtexture-on-the inside-of-a-sphere.html

Sounds

- Water audio
  https://www.youtube.com/watch?v=YtQK38eyqKU

Appendix I

Code

I.1 Movement1: Teleport code

```
//======= Aalborg University 19gr1041 ================================
//
// Name: Teleport To DropOff With Bazier Curve
//
// Purpose: This script uses the same functionalities as the standart
    ↪ teleport
// script, however as a trajectory it will use a bezier curve to draw
// the path and to translate along, it does not need a pathviewer
//
//====================================================================

using UnityEngine;
using Valve.VR.InteractionSystem;
using Valve.VR;

/// <summary>
/// Purpose: This script uses the same functionalities as the standart
    ↪  teleport
/// script, however as a trajectory it will use a bezier curve to draw
/// the path and to translate along, it does not need a pathviewer.
/// </summary>
public class Movement2_Bazier : MonoBehaviour
{
    public SteamVR_Action_Boolean teleportAction = SteamVR_Input.
        ↪ GetAction<SteamVR_Action_Boolean>("Teleport");
    [SerializeField] public TeleportArc tpArc;
```

```
[SerializeField] private LineRenderer _lr;
private float amountOfPoints = 100.0f;
[SerializeField] public float stepper = 1.0f;
public bool _playerIsMoving = false;
[SerializeField] private MeshRenderer _targetMeshR;

float height, lengthOfJurney, time, test;
Vector3 goalPoint, startPoint;
bool go, draw, isAlowedToGoThere, redraw;
Vector3[] points, fixedPointsFOrLinerender;
int index = 0;

private static Movement2_Bazier _instance;
public static Movement2_Bazier instance
{
    get
    {
        if (_instance == null)
        {
            _instance = FindObjectOfType<Movement2_Bazier>();
        }
        return _instance;
    }
}

void Awake()
{
    if (this.enabled)
    {
        _lr.enabled = true;
        Teleport teleportScript = tpArc.gameObject.GetComponent<
            ↪ Teleport>();
        teleportScript.enabled = true;
    }
}

void Start()
{
    startPoint = Player.instance.transform.position;
    height = startPoint.y;
    _lr.material.SetTextureScale("_MainTex", new Vector2(1, 1.0f));
```

```
        _lr.positionCount = (int)amountOfPoints + 1;
    }

    // Update is called once per frame
    void Update()
    {
        if (_playerIsMoving && Player.instance.transform.position ==
            ↪ goalPoint)
        {
            _playerIsMoving = false;
            //Debug.Log("It took me " + (Time.time - test) + "seconds to
                ↪  get here!");
        }
        else if(_playerIsMoving)
        {
            _targetMeshR.enabled = false;
        }

        if (tpArc.collisionName != "nothing")
        {

            isAlowedToGoThere = true;
        }
        else
        {
            isAlowedToGoThere = false;
        }


        if (Input.GetKey(KeyCode.T) && !_playerIsMoving ||
            ↪ IsTrueHandsGetKey() && !_playerIsMoving)
        {
            if (!_lr.enabled)
            {
                _lr.enabled = true;
            }

            goalPoint = tpArc.EndPosition();
            lengthOfJurney = Vector3.Distance(Player.instance.transform.
                ↪ position, goalPoint);
            draw = true;
```

```
    }

    if (draw && !redraw)
    {
        points = new Vector3[(int)amountOfPoints + 1];

        for (int i = 0; i < (int)amountOfPoints; i++)
        {
            points[i] = BazierPoint(startPoint, goalPoint, (1 /
                ↪ amountOfPoints) * i);
            points[100] = goalPoint;
        }

        _lr.SetPositions(points);
    }



    if (Input.GetKeyUp(KeyCode.T) && isAlowedToGoThere ||
        ↪ IsTrueHandsGetKeyUp() && isAlowedToGoThere)
    {
        go = true;
        _playerIsMoving = true;
        time = Time.time;
        test = Time.time;

    }

    if (go)
    {
        redraw = true;
        //Debug.Log("points.length: " + points.Length + " index: " +
            ↪  index);


        if (index <= points.Length-1)
        {
            Player.instance.transform.position = Vector3.MoveTowards
                ↪ (Player.instance.transform.position, points[index
                ↪ ], 100.0f * Time.deltaTime * SlowerWhenClose.
```

```
                ↪ instance.Speed());
            if (Player.instance.transform.position == points[index])
            {
                for (int i = 0; i <= index; i++)
                {
                    points[i] = Player.instance.feetPositionGuess;
                    _lr.SetPositions(points);
                }
                index++;
            }
        }
        else
        {
            _lr.positionCount = 0;
        }
    }
}

private bool IsTrueHandsGetKey()
{
    bool bla;
    if (UnityEngine.XR.XRDevice.isPresent)
    {
        if (teleportAction.GetState(Player.instance.rightHand.
            ↪ handType))
        {
            bla = true;
        }
        else
        {
            bla = false;
        }
    }
    else
    {
        bla = false;
    }

    return bla;
}
```

```
private bool IsTrueHandsGetKeyUp()
{
    bool bla;
    if (UnityEngine.XR.XRDevice.isPresent)
    {
        if (teleportAction.GetStateUp(Player.instance.rightHand.
            ↪ handType))
        {
            bla = true;
        }
        else
        {
            bla = false;
        }
    }
    else
    {
        bla = false;
    }

    return bla;
}

private Vector3 BazierPoint(Vector3 currentPos, Vector3 goalPos,
    ↪ float percentToGo)
{
    Vector3 currentStep;
    Vector3 abovePoint;
    float pCurrentX;
    float pCurrentY;
    float pCurrentZ;

    abovePoint = new Vector3(goalPos.x, height, goalPos.z);
    pCurrentX = Mathf.Pow(1 - percentToGo, 2) * currentPos.x + (1 -
        ↪  percentToGo) * 2 * percentToGo * abovePoint.x +
        ↪ percentToGo * percentToGo * goalPos.x;
    pCurrentY = Mathf.Pow(1 - percentToGo, 2) * currentPos.y + (1 -
        ↪  percentToGo) * 2 * percentToGo * abovePoint.y +
        ↪ percentToGo * percentToGo * goalPos.y;
    pCurrentZ = Mathf.Pow(1 - percentToGo, 2) * currentPos.z + (1 -
        ↪  percentToGo) * 2 * percentToGo * abovePoint.z +
```

```
            ↪ percentToGo * percentToGo * goalPos.z;

        currentStep = new Vector3(pCurrentX, pCurrentY, pCurrentZ);
        return currentStep;
    }
}
```

I.2 Movement2: Controller code

```
//======= Aalborg University 19gr1041 ================================
//
// Name: Pedal controller FreeLook
//
// Purpose: this script if enabled lets the the VR player use the
    ↪ controllers
//
//=====================================================================

using UnityEngine;
using Valve.VR;
using Valve.VR.InteractionSystem;

public class Movement_3 : MonoBehaviour
{
    [SerializeField] private bool _dynamicMovementSpeed = false;
    private float _speed = 1.0f;
    private float _speedKonstantBack = 0.25f;
    private float _speedKonstant = 0.02f;

    public SteamVR_Action_Boolean startMovementAction = SteamVR_Input.
        ↪ GetAction<SteamVR_Action_Boolean>("startMovement");
    private bool _driveOn;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
```

```
{
    if (startMovementAction.GetState(Player.instance.rightHand.
        ↪ handType))
    {
        _driveOn = true;
    }
    else
    {
        _driveOn = false;
    }

    if ( RightForward() && _driveOn)
    {
        _speed = Player.instance.rightHand.transform.rotation.
            ↪ eulerAngles.x;

        if (_dynamicMovementSpeed)
        {
            Player.instance.transform.position += Player.instance.
                ↪ rightHand.transform.localRotation * Vector3.
                ↪ forward;
        }
        else
        {
            Player.instance.transform.position += Player.instance.
                ↪ rightHand.transform.localRotation * Vector3.
                ↪ forward * SlowerWhenClose.instance.Speed();
        }



    }
    else if ( RightBackwards() && _driveOn)
    {
        _speed = Player.instance.rightHand.transform.rotation.
            ↪ eulerAngles.x;

        if (_dynamicMovementSpeed)
        {
            Player.instance.transform.position -= Player.instance.
                ↪ rightHand.transform.localRotation * Vector3.back;
```

```
        }
        else
        {
            Player.instance.transform.position -= Player.instance.
                ↪ rightHand.transform.localRotation * Vector3.back
                ↪ * SlowerWhenClose.instance.Speed();
        }

    }


}


//350x is horizontal ish

//320 is lifting it for backwards

//30 is tilting down forward



private bool RightBackwards()
{
    bool returnValue;

    if (Player.instance.rightHand.transform.rotation.eulerAngles.x
        ↪ <= 359f && Player.instance.rightHand.transform.rotation.
        ↪ eulerAngles.x >= 270f)
    {
        returnValue = true;
    }
    else
    {
        returnValue = false;
    }

    return returnValue;
}

private bool RightForward()
```

```
    {
        bool returnValue;

        if (Player.instance.rightHand.transform.rotation.eulerAngles.x
            ↪ >= 1f && Player.instance.rightHand.transform.rotation.
            ↪ eulerAngles.x <= 90f)
        {
            returnValue = true;
        }
        else
        {
            returnValue = false;
        }

        return returnValue;
    }
}
```

I.3 Movement3: Trackpad code

```
//======= Aalborg University 19gr1041 ===============================
//
// Name: Trackpad movement
//
// Purpose: move the VRPlayer according to trackpad input
//
//===================================================================

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Valve.VR;
using Valve.VR.InteractionSystem;


public class Movement_5 : MonoBehaviour
{
    [SerializeField] private bool _hmdRot;
    public SteamVR_Action_Vector2 touchPadAction;
    private float _speed;
```

```
    void Update()
    {
        //======= The Touchpad
            ↪ ========================================
        // Unit Circle
        // 1
        // |
        // |
        // -1------|------1
        // |
        // |
        // -1
        //===============================================================
            ↪

        Vector2 touchpadValue = touchPadAction.GetAxis(
            ↪ SteamVR_Input_Sources.Any);

        if (touchpadValue != Vector2.zero && !_hmdRot)
        {
            _speed = Mathf.Abs(Vector2.Distance(touchpadValue, Vector2.
                ↪ zero)) * SlowerWhenClose.instance.Speed() * 2;
            Player.instance.transform.position += Player.instance.
                ↪ rightHand.transform.rotation * (new Vector3(
                ↪ touchpadValue.x, 0f, touchpadValue.y) *_speed);

        }
        else if (touchpadValue != Vector2.zero && _hmdRot)
        {
            _speed = Mathf.Abs(Vector2.Distance(touchpadValue, Vector2.
                ↪ zero)) * SlowerWhenClose.instance.Speed() * 2;
            Player.instance.transform.position += Player.instance.
                ↪ hmdTransform.rotation * (new Vector3(touchpadValue.x,
                ↪  0f, touchpadValue.y) * _speed);
        }
    }
}
```

I.4 Movement4: AutoMove code

```
//======= Aalborg University 19gr1041 ================================
//
// Name: Auto move
//
// Purpose: Move the player automatically
//
//====================================================================


using UnityEngine;
using Valve.VR.InteractionSystem;
using UnityEngine.UI;


public class AutoMove : MonoBehaviour
{
    [SerializeField] private float _pushSpeed;
    [SerializeField] private float _whenFade = 30.0f;
    [SerializeField] private Image _vrFade;
    [SerializeField] private Image _fallbackFade;
    [SerializeField] private Vector3 _goalPosition = new Vector3(204f,
        ↪ 8.6f, 111f);
    private Vector3 _startPos;
    private Rigidbody _rb;
    private float _startTime, startTime;
    private float journeyLength;

    void Start()
    {
        _rb = Player.instance.gameObject.AddComponent<Rigidbody>();
        _rb.useGravity = false;
        _startPos = Player.instance.transform.position;
        journeyLength = Vector3.Distance(_startPos, _goalPosition);
        startTime = Time.time;
    }

    void Update()
    {
        float distCovered = (Time.time - startTime);
```

```
        float fracJourney = distCovered / journeyLength;

        Player.instance.transform.position = Vector3.Lerp(Player.
            ↪ instance.transform.position, _goalPosition, fracJourney)
            ↪ ;
        if (_startTime == 0.0f)
        {
            _startTime = Time.time;
        }
        if (Time.time - _startTime >= _whenFade)
        {
            _vrFade.color = new Color(0,0,0, 0);
            _fallbackFade.color = new Color(0, 0, 0, ((Time.time -
                ↪ _startTime) - _whenFade + 1.5f) * 0.5f);
        }
    }
}
```

I.5 Dynamic speed adaption

```
//======= Aalborg University 19gr1041 ================================
//
// Name: Slower When Close
//
// Purpose: this script will check distances (all 6 directions)
// if a distance is to short, it will start fading it out by the
    ↪ amount your
// are away from a surface minus a constant. This results in being
    ↪ pitch black
// when to close to surfaces preventing impact feeling
// In addition it will throw a value that is used by all movement
    ↪ methods in
// the worldview scene, resulting in addaptive speed
// (Slower when close to surfaces)
//===================================================================

using System;
using UnityEngine;
using UnityEngine.UI;
using Valve.VR.InteractionSystem;
```

```
/// <summary>
/// Purpose: this script will check distances (all 6 directions)
/// if a distance is to short, it will start fading it out by the
    ↪ amount your
/// are away from a surface minus a constant. This results in being
    ↪ pitch black
/// when to close to surfaces preventing impact feeling
/// In addition it will throw a value that is used by all movement
    ↪ methods in
/// the worldview scene, resulting in addaptive speed
/// (Slower when close to surfaces)
/// </summary>
public class SlowerWhenClose : MonoBehaviour
{
    [SerializeField] private Image _fadeOut;
    [HideInInspector] public bool fadedOut = false;
    private LayerMask layerMask;
    private float _distance, _relativeDistance;

    private static SlowerWhenClose _instance;
    /// <summary>
    /// Gets the instance of the SlowerWhenClose script, this is used
        ↪ to get the adaptive speed.
    /// </summary>
    /// <value>The instance.</value>
    public static SlowerWhenClose instance
    {
        get
        {
            if (_instance == null)
            {
                _instance = FindObjectOfType<SlowerWhenClose>();
            }
            return _instance;
        }
    }

    void Start()
    {
        layerMask = ~(1 << LayerMask.NameToLayer("IgnoredByTeleport"));
            ↪ // ignore collisions with the objects that have the
```

```
        ↪ layer set to: IgnoredByTeleport

}

void Update()
{
    float leftDistance = RayCastDistance(Vector3.left);
    float rightDistance = RayCastDistance(Vector3.right);
    float upDistance = RayCastDistance(Vector3.up);
    float downDistance = RayCastDistance(Vector3.down);
    float frontDistance = RayCastDistance(Vector3.forward);
    float backDistance = RayCastDistance(Vector3.back);

    var rayCastArray = new[] { leftDistance, rightDistance,
        ↪ upDistance, downDistance, frontDistance, backDistance};
    Array.Sort(rayCastArray);

    for (int i = 0; i <= rayCastArray.Length-1; i++)
    {
        if(rayCastArray[i] != 0.0f)
        {
            _distance = rayCastArray[i];
            if (_distance < 5f && _distance != 0.0f)
            {
                if (_fadeOut.transform.parent.gameObject.activeSelf
                    ↪ == false)
                {
                    _fadeOut.transform.parent.gameObject.SetActive(
                        ↪ true);
                }

                if (((5 - _distance) * 0.34f) >= 1.0f)
                {
                    fadedOut = true;
                    _fadeOut.color = Color.black;
                    return;
                }
                else
                {
                    fadedOut = false;
                }
```

```
                _fadeOut.color = new Color(_fadeOut.color.r,
                    ↪ _fadeOut.color.g, _fadeOut.color.b, (5 -
                    ↪ _distance) * 0.34f);
            }
            else
            {
                _fadeOut.transform.parent.gameObject.SetActive(false
                    ↪ );
            }
            return;
        }
        else
        {
            _fadeOut.transform.parent.gameObject.SetActive(false);
        }
    }
}

private float RayCastDistance(Vector3 axisToUse)
{
    RaycastHit rayCastHit;
    float rayLenght;
    rayLenght = 0.0f;

    Ray rayCast = new Ray(Player.instance.hmdTransform.position,
        ↪ Player.instance.hmdTransform.rotation * axisToUse);

    if (Physics.Raycast(rayCast, out rayCastHit, Mathf.Infinity,
        ↪ layerMask))
    {
        rayLenght = rayCastHit.distance;
    }

    return rayLenght;
}

/// <summary>
/// A value between 0 and 1 for adapting the speed of movement
    ↪ methods,
```

```
    /// the return value is dependent on the distance to the nearest
        ↪ colliders of the VRPlayer
    /// </summary>
    public float Speed()
    {
        return 0.3f + (1 - 0.3f) * ((_distance - 2) / (500 - 2));
    }
}
```