Redirected walking and VR-sickness

Rahel Jamal Gaeb

Oppgave for graden
Master i Informatics: Programming and System Architecture
60 studiepoeng

Institute for Informatics
Det matematisk-naturvitenskapelige fakultet

UNIVERSITETET I OSLO

Høsten 2023

Redirected walking and VR-sickness

Rahel Jamal Gaeb

Redirected walking and VR-sickness

http://www.duo.uio.no/

Trykk: Reprosentralen, Universitetet i Oslo

Abstract

Redirected walking is a technique used to extend the physical play space of a virtual reality implementation through manipulating the senses of the user. Techniques such as amplifying the rotation, the movement and tricking the user into correcting their path by subtly rotating. Such methods may turn a small play-space into a larger space, or a large space into a possibly infinite space. Such benefits are not without their cost. VR headsets can strain and cause unpleasant effects on the user by itself. Redirecting the user can further amplify this effect as the vestibular system, which helps the body maintain its balance through fluids passing through fine hairs in circular ducts will not be fooled unlike the eyes. Mismatches can cause a condition called virtual reality sickness, a set of symptoms similar to motion sickness, but arise when the sensory organs have conflicting stimuli.

A Standardized approach is used through a simulator sickness questionnaire (SSQ) which has been modified for use on VR use-cases. The higher the redirection intensity, the greater the play-space can be extended, but the greater the sickness score. Factors such as redirection type and intensity are important to look at in order to determine how people react to them. The goal of this thesis is to understand what technical factors in a redirected walking system contribute to negative effects. The methodology chosen for the study is a mix of quantitative and qualitative methods alongside a gamified VR experience, while they are being exposed to redirection algorithms. There were no statistically significant differences, however the translation algorithm produced a 44.4% higher score than the lowest algorithm, and a 22% higher score than the second highest algorithm.

All intensity level showed a positive trend. Higher scales lead to higher sickness scores. The curvature algorithm shows potential considering the growth per scale, which represents the radius of a circle for use in smaller spaces. No conclusion was able to be drawn due to lack of statistical significance.

Acknowledgements

I would like thank my supervisor Carsten Griwodz for all his scientific insight during the writing of this thesis, and in addition his course IN5060 - Performance in distributed systems. Lessons learned from the course relating in statistics and analysis proved to be significantly helpful in analyzing the data produced from this thesis.

I am also grateful for the University of Oslo SINLAB who provided the virtual reality headset and computing hardware necessary to develop and deploy the Redirected Walking algorithm

I would also like to thank Halvor Kristian Ringsby and Emil Magnar Kjenstad for their exceptional contribution in developing and designing the Redirected Walking implementation and their suggestions into how handle and analyze the data from the Redirected Walking system.

Lastly, I would like to thank my family, who motivated me through tiresome and long nights of reading, writing and testing.

Contents

1 Introduction

1.1 Motivation

Imagine being able to wear a pair of virtual reality googles. That you can imagine, however most likely you probably wouldn't dare stand up with them, or could you? But what if you were to hit a wall, surely this is unavoidable, but the story could easily fare in a different direction, both your direction and the story of having to sit down to use your virtual reality headset. The idea isn't mere imagination, nor truth, it is something in the middle. It is called Redirected Walking. But as the saying goes, everything that glitters is not gold. In this case it is still shiny and valuable. It just needs a bit of polish.

The concept of redirected walking, in its broadest strokes, encompasses different techniques to redirect the user into walking a path that they themselves do not realize they are walking in. Say as an example a user wishes to walk in a straight line, however this means that the user might collide with a wall assuming an enclosed space. Normally this would not be an issue, as the user can observe the boundaries of their space and avoid the wall, however this is not the case when wearing a virtual reality headset.

Traditionally, in VR, there exists several ways for the user to give movement input to the computer. VR headsets include hand held controllers, typically a pair of grip like controllers with buttons and perhaps thumb-sticks. The user sits down in a chair and gives input through these. These controllers might also have their position tracked by the headset, and allow the actual movement of the controllers themselves to give input. One such application might be to move a set of hands in the VR world, which is done in a more natural way by the actual movement of the controllers versus using the thumb-sticks. The user might also drop the use of the controllers and instead use a keyboard, assuming they have the muscle memory to do so without looking at the keyboard.

One more modality exists, in which the position of the headset, by extension the position of the user is used as input to the VR headset. The user can look around by tilting their head, rotating laterally or horizontally to look around in the VR world. This can be extended upon by including the actual movement of the user in the real-space to the virtual reality space. In this way the user immersion could be improved by letting the user explore a virtual world by the means of their actual physical locomotion.

An issue arises when the user is wearing a virtual reality headset. Such a device would block the users line of sight, and a collision would easily occur. Even in an completely open space, a user might feel anxious to move, even if they know there is nothing to collide with. The solution for this problem would be to employ a redirected walking approach. Instead of the user taking a collision course, the redirected walking algorithm would redirect the user out of such a collision course.

It is critical for virtual reality applications to try to tackle this issue if they wish to get an immersive experience for the user. Not every virtual reality headset owner has access to a very large space to navigate in. Perhaps they might use their living room cleared of obstacles to enjoy the experience VR provides, however not every user might have access to a living room of sufficient size, nor the motivation to move couches, dinner tables and other furniture out of the way. Student accommodations, small apartments and the placement of support hardware; such as a computer with discrete graphics processor limit the areas of which immersive VR with locomotion might be employed. It is therefore critical to tackle this issue in context of the free space required, and even the computational complexity of the algorithm.

Even if the issues are tackled, then arises the issue of how the human body interacts with the VR-headset. Humans do not enjoy having their senses manipulated, and many issues arise when the body receives stimuli which is contradictory to other sensory organs. In these cases, VR-sickness can arise, a sickness with similar symptoms to motion sickness. The redirection could be perfect, however nausea, eyestrain and other symptoms which compose VR-sickness could "pull the plug" on redirected walking if it is not tackled adequately

Lastly, if such an technology is to make its way into the hands of corporations, conglomerates and companies, then it must be bulletproof when it comes to sickness. Imagine an employee who becomes sick and unwell after such an experience. It could spell litigation, fines and inspections from HSE (Health, Safety and Environment) organisations. In-fact most countries have laws that mandate the employer to provide a safe physical working environment.[1] As such, if redirected walking wishes its adoption by the large gears of society, they must prove themselves to be a safe *and* effective technology.

1.2 Research Questions

The goal of this thesis is to identify technical factors in an redirected walking algorithm that contribute to "VR-sickness", a type of sickness that is closely related to simulator sickness on the basis of symptom overlap.

Technical factors in an redirected walking algorithm involve the strength of the redirection, which typically is referred to as the "gain". In addition, the type of algorithm itself is a technical factor, such as the choice of translation, curvature or rotational algorithm.

- **RQ1:** How does the scale parameter in an redirected walking algorithm influence the prevalence of VR-sickness symptoms in an redirected walking setup?

---

[1] Arbeids- og inkluderingsdepartementet. *Lov om arbeidsmiljø, arbeidstid og stillingsvern mv. (arbeidsmiljøloven).*
`https://lovdata.no/lov/2005-06-17-62/4-1`. 2022.

- **RQ2:** How does the choice of redirection technique affect the prevalence of VR-sickness symptoms in an redirected walking setup?
- **RQ3:** What factor or combination of factors cause VR-sickness in redirected walking setups?

In addition to the research questions, a hypothesis is formulated.

- **Hypothesis**: The scale (See section 3.7) parameter is the only factor that produces a statistically significant increase in VR-sickness symptoms. The choice of redirection algorithms: rotation (section 3.7.1), translation (3.7.2 and curvature (3.7.3 do not impact the frequency of vr-sickness symptoms

1.3 Contributions

The research questions listed in the earlier section represent important aspects that must be addressed in order for redirected walking to a trusted and well known technology. The paper and the research questions have the potential to contribute to the following points

- **Contribution:** The understanding of how to minimize sickness risk when it comes to stand-up virtual reality experiences with redirected walking.
- **Contribution:** The ability of corporations, institutions and individuals who wish to employ redirected walking alongside virtual reality to understand the risks and benefits.
- **Contribution:** Allow the aforementioned groups to accurately weight the use against legislation and ethical concerns using research and not "gut-feeling" or conjecture

The three aforementioned points represent the condensation of several smaller factors that arise when considering redirected walking and virtual reality.

Such factors are divided into ethical, legal and technical factors. this thesis has the potential to contribute, and perhaps explain these points.

2 Background research

2.1 Virtual Reality Headsets

Virtual reality headsets, also known as "VR-Headsets" are head mounted devices that provide the user with an virtual reality experience. Such a device contains a screen mounted behind a lens, or in more advanced consumer packages two lenses which combined provide an stereoscopic view to the user so that it seems as if they are observing a realistic scene featuring an illusion of depth.

Some headsets also include hand-held controllers shaped in such a way that they are to be used in a hand each. The user holds a controller in each hand, and the headset tracks and displays the controllers as hands and projects them on the screen. This gives the user the illusion of having their hands in the virtual world, despite being obstructed by the bulk of the headset. This is commonly referred to as "hand tracking" however other means of tracking the hands are possible, such as camera based tracking, gloves - which may provide tracking of each digit.

These controllers might employ vibration feedback, as to give a more immersion. When the users hands come in contact with an object in the virtual world they start to vibrate. Other methods might include the use of ultrasound fields to give some feedback to the user, through the use of air pressure or the use of motors to either pull or push the user's digits to emulate the sensation of force when an object is held. These techniques generally go under the term "Haptic Feedback". They try to generate the feeling of contact with objects and VR headsets employ these, typically simple vibration, to indicate contact with an object.

In addition to the aforementioned aspects, sound is also an element virtual reality headsets use, specifically stereo sound. In the context of redirected walking, sounds may be employed to distract the user. A heavy or unexpected sound may be played, but its origin being perhaps behind the user. The user will hear the sound, and turn towards the sound source to investigate what caused the sound.

In order to immerse the user even more, it may be necessary to insulate the user from real life noise, so that they cannot locate themselves in the room by hearing the sound. In this scenario, the employment of noise cancelling techniques could be used to counteract the ambient noise. An easier solution to achieve this is to simply play a slightly louder ambient sound such as ocean waves or forest noises in an effort to drown out the ambient sound. This approach could be combined with closed back headphones which could mean that the effect could be achieved with a lower decibel sound.

2.2 Redirected walking techniques

Considering the limited spaces in which VR headsets that track the users movements as a way to navigate may be employed, the need for redirection arises. There are a few redirection techniques that could make VR in such enclosed spaces possible, however the downsides are that some of them require a minimum radius in order to navigate the user in a space with no chance of collision. Different techniques exist, but most of the techniques rely on subtly manipulating the input that the user gives, or subtly manipulating the scene that the user observes. In essence, the user is being redirected through these manipulations towards a desired path.

2.3 One-to-N mapping (Translation)

One technique is to simply map displacement in the real world by a factor N in the digital world. In practical terms an example of this assuming $N = 2$

would mean that a displacement of 1 meter in the physical world would map into a displacement of 2 meters in the digital world. Assuming a room with the dimensions of two by two meters alongside $N = 2$, the user would be able to move in a four by four virtual square space. This is not a very flexible solution, as the physical space constraints are still met if the user has to move in a straight line in the virtual world. Also, it may be difficult to do precise movement is the N factor is high enough.

One could increase the value N however the disconnect between perceived movement, and actual movement will grow. When this disconnect grows, the user might be afflicted with motion sickness caused by the disconnect between perceived displacement and actual displacement as experienced by the canals in the ear.[2] If we assume that the user will not experience motion sickness, there still is the issue of space, as constant movement in a direction will lead into a wall, of which the user will be reminded by a physical impact with possible minor trauma.

The solution to the issue above can possibly be solved by using on-board sensors in the VR headset. Most VR headsets feature a wide variety of sensors that the headset uses to position the user in the digital world. Initially, the user has to designate a "safety" grid in which they can navigate. Once the user is too close to the safety grid, the head mounted display (HMD) will instead show the user the real scene via cameras mounted on the VR headset frame. This is usually accomplished by employing a SLAM (Simultaneous localization and mapping) algorithm to determine if the user is too close to the specified safety boundaries.

Some headsets eschew this approach and use a simple time-of-flight sensor (TOF). Once the sensor reports that the distance between the headset and measured target is close to some safety parameter, the HMD will display the

---

[2] Joseph J. LaViola. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32 (1 Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: `10.1145/333329.333344`.

real scene.

One-to-N mapping could be combined with the aforementioned techniques, such as the safety grind mechanism; in which the real scene is displayed if the user has passed the safety grid, or is about to pass the grid, or the ToF sensor detecting a wall in which again, will display the real world scene, or even give a small warning on the user's interface.

With the techniques above, One-to-N mapping could be used, however the major caveat with these approaches is that immersion will be broken frequently if one wishes to move in a straight line. In order to create a truly immersion experience, the user has to not be aware of their surroundings, and instead completely believe in the stimuli the VR headset is providing them.

2.4 Steering algorithms

Steering algorithms are a family of algorithms that attempts to steer the user away from the boundaries of the VR play space. In order to achieve this, the algorithm picks a destination for the user, and attempts to rotate the view that the user sees, so that the correction applied by the user will steer them away from the boundary and towards some specified point. Variants of this steering algorithm exist, such as Steer-to-center; a variant that always attempts to steer the user towards the center of the boundary, steer-to-orbit; a variant that tries to make the user "orbit" a point by navigating the user towards the point, then tries to path the user around the point.

The upside of such "steering" algorithms is that if done properly, they will allow the user to navigate a virtual world for a quite some time, perhaps until the user has is too tired from wearing the headset. The steering algorithm attempts to ensure that the user is a safe distance away from the boundaries of the VR world.

The Steer-to-center variant is an easier algorithm to implement from a conceptual standpoint. The algorithm requires the center point of the play space, and the resulting paths that the user takes curve away from the user. This results in the user performing many micro-adjustments to the path, which given a large enough play-space will cause the user to pass through the center point. Subsequent movement would try to make the user pass through the center again using the same strategy described above.

Steer-to-orbit is another variant of this algorithm. Imagine a satellite; the Moon in orbit around Earth. If the trajectory of the Moon was drawn, then it would show a path that would "orbit" the centre point, or a path that would steer the user towards orbit. The "user" orbits a selected point in the real play-space. The advantage of this algorithm is that the room can be in a rectangular shape, this would simply produce an elliptical orbit. An elliptical orbit might produce at times lower rotation in degrees per second, but as the user approaches the vertex points, large rotations would be produced. This could perhaps break immersion due to how harsh the rotation will be.

Constant rotation solutions would perhaps produce fewer "spikes" in rotations speed, which could aid in the comfort aspect of the VR session. This might mean that steer-to-centre or steer-to-orbit (where rotation speed is constant) could be used to create a better redirected walking implementation for virtual reality.

The "blinking" aspect of the rotation might not be sufficient for every redirected walking implementation. Assume that the blinking mechanism is the only method used. If a user approaches a boundary at 90 degrees relative to the obstacle, and the algorithm engages the blink mechanism and rotates the view 9.1 degrees, then further blinks will be needed, and this would result in many blinks that could disorient the user. In the Nguyen and Kunz paper[3],

---

[3] Anh Nguyen and Andreas Kunz. "Discrete scene rotation during blinks and its effect on redirected walking algorithms". In: ACM, Nov. 2018, pp. 1–10. ISBN: 9781450360869. DOI: `10.1145/3281505.3281515`.

they combined the blink mechanism with classic rotation based redirected walking algorithms.

Another type of algorithm is curvature based. The goal of the curvature based algorithms is to curve the path that the user is following to direct them to either side. Suppose the user is walking towards some goal in-front of them, and walks straight, then the virtual world is rotated such that the user performs a correction without realizing it. This creates a curved path, and can lead the user back into orbit (if steer-to-orbit is used) or towards the center (in the case of steer-to-center algorithm). The curvature algorithm can be implemented using the auxiliary techniques (explained in the next section) with physical feedback (visuo-haptic feedback) and has been showcased in earlier research.[4]

There is however a major issue with such algorithms, the space required to create the curved paths that steer the user set certain space requirements for the play-space. One paper established a 22 meter radius of the arced path for an implementation that slightly rotates the user's view to steer them.[5] This implies a room with a minimum requirement far surpassing most households. With such an implementation, the user would have to rent a sports hall simply to enjoy immersion VR walking, or enjoy the experience outside in a park or other suitable spaces. There have been advances in steering algorithms to allow for much smaller playspaces, the downside again being that pre-defined paths have to be specified. This reduces the playspace down to a $36m^2$, and allows the paths to curve up to twenty degrees without the users noticing.[6]

---

[4] Hiroaki Sakono et al. "Redirected Walking using Continuous Curvature Manipulation". In: *IEEE Transactions on Visualization and Computer Graphics* 27.11 (2021), pp. 4278–4288. DOI: `10.1109/TVCG.2021.3106501`.

[5] Eike Langbehn, Paul Lubos, and Frank Steinicke. "Evaluation of Locomotion Techniques for Room-Scale VR". in: ACM, Apr. 2018, pp. 1–9. ISBN: 9781450353816. DOI: `10.1145/3234253.3234291`.

[6] Michael Rietzler et al. "Rethinking Redirected Walking: On the Use of Curvature Gains Beyond Perceptual Limitations and Revisiting Bending Gains". In: IEEE, Oct.

2.5 Auxiliary Techniques

The sections below describe techniques which by themselves may not redirect the user completely, but can be combined with other redirected walking algorithms such as translation or rotation can augment the effectiveness of the primary algorithm used. They can be used by themselves, but may limit the modality of the vr-application. Regardless, they are important to mention due to their ability to redirect, regardless of their efficacy.

2.6 Acoustic Manipulation

Acoustic manipulation is a redirection technique in which sound in a 3D space is presented to the user in such a way to distract them and make the user turn towards the sound origin or even away from it.[7] In this way, the algorithm can steer the user away from physical boundaries or in tandem with other redirected walking techniques to enhance their redirection. In order to use such manipulations, the VR headset must be equipped with headphones capable of stereo sound. The stereo sound system is used to play noises that the user's ears will locate in 3D space, based on the origin and intensity. This is not possible with a mono sound source, as the user will not be able to discern where the sound is coming form, only its intensity.

2.7 Scene rotation

The user is presented with the VR world through an HMD. One way to implement redirected walking through manipulation of what the user sees is typically called scene rotation. Such rotating approaches either rotate the digital world itself, or simply the user's field of view. For example, when the user is too close to a wall, then the HMD can rotate the displayed scene

---

2018, pp. 115–122. ISBN: 978-1-5386-7459-8. DOI: `10.1109/ISMAR.2018.00041`.

[7] Tobias Feigl et al. "Acoustical manipulation for redirected walking". In: ACM, Nov. 2017, pp. 1–2. ISBN: 9781450355483. DOI: `10.1145/3139131.3141205`.

to either the left, or to the right. The user then has to reorient themselves and if done correctly, they will move towards their intended target without maintaining a course in the physical world that would result in a collision. One issue with such an approach is having to balance the rotation in such a way that the user does not realize it has occurred, but also in a way that makes it effective in avoiding collisions.

Findings in articles seem to lean towards that such scene rotations are possible without the user noticing; from either observations or motion sickness. In one article, it was found that a rotating the scene 2.4 degrees was not noticeable for the user. If the screen temporarily "blacked" emulating the natural process of blinking, then a rotation of 9.1 degrees was possible.[8]

By combining several of the techniques described above, it may be possible to create a more accessible redirected walking environment for the end users. Currently the only factor that is important is the size of the play-space limiting the deployment of such algorithms. However, this is not the case. While the algorithms are described alongside the space requirements, it is important to include the user into this question. The user could be young or old, sedentary or active, they may have very good balance or stumble easily. The point is that not every user can handle the same rotation or misdirection of their senses.

Dancers might handle rotation worse. Some may be more easily "redirected" by their lack of motion senses, while some might see completely through it due to being more in touch with their motion senses. Some might even have a negative reaction to the redirection; such as dizziness, emesis (throwing up) or loss of coordination. It is important to map these factors alongside the algorithms that are to be employed, to chose the best experience for the user while ensuring their safety.

---

[8] Anh Nguyen and Andreas Kunz. "Discrete scene rotation during blinks and its effect on redirected walking algorithms". In: ACM, Nov. 2018, pp. 1–10. ISBN: 9781450360869. DOI: `10.1145/3281505.3281515`.

2.8 Motion sickness and redirected walking

As mentioned earlier, we are in principle trying to trick the users senses. This can be problematic, as VR headsets cannot trick all senses of the body, at least easily. One problem that arises is motion sickness.

Motion sickness is typically attributed due to the mismatch between perceived motion and the vestibular system; a system which provides the brain with information about the body's orientation, movement and spatial information. The vestibular system is approximately located behind the ear drum. There exists many theories as to why this occurs when the vestibular system is disturbed.

One theory, which can be applied to the situation of redirecting the user, is the "sensory conflict" theory. Simply described, if there is a difference between perceived motion, and the motion that the vestibular system experiences, then motion sickness may occur.[9] This fits in with virtual reality and could be postulated to become an ever bigger aspect when considering the use of redirected walking algorithms. The user is constantly given visual information about the virtual environment that they are observing, however sensory input from other organs, such as the ears or physical contact might give conflicting sensory data, and as such might cause motion sickness.

Many factors are documented which can cause motion sickness. These are not necessary conditions, but only sufficient conditions. They may be present, but their presence does not alone create motion sickness. Individuals may get motion sick without these factors being present. One such example is imperceivable acceleration, which by itself can cause motion sickness. It is important to design the redirected walking algorithm in such a way to minimize these factors if possible in order to create a more user friendly

---

[9] Joseph J. LaViola. "A discussion of cybersickness in virtual environments". In: *ACM SIGCHI Bulletin* 32 (1 Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: `10.1145/333329.333344`.

implementation

Closely related to motion sickness is ”cyber-sickness”. Cyber sickness is sickness that typically causes symptoms such as eye-strain, headache, disorientation and emesis. These symptoms, amongst other not listed, are shared by motion sickness, however cyber sickness can occur when the user is not moving at all, as opposed to motion sickness, which typically occurs due to the sensory mismatch described above.[10]

This condition can even occur after the user has finished with a virtual reality experience.[11] Imagine swimming for an hour, only to leave the pool and go on with your day. You might experience a disconnect with your weight and the movement. This same disconnect could apply the following hours after a redirected walking experience. It is estimated that somewhere between 50 to 80 percent of users will be affected by some sort of symptom that relate to cyber-sickness, such as nausea, fatigue or loss of coordination.[12]. An older study from the 1995 found that 61 percent of users (146 participants) will suffer from negative effects after 20 minutes in a VR simulator.[13] These metrics relate to sit-down VR without redirected walking.

It is possible to predict how susceptible a user will be to cyber sickness, however the study supporting this found the correlation between certain factors (e.g. sex, health, fitness and age) and VR exposure, but not necessarily ex-

---

[10] Joseph J. LaViola. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32 (1 Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: `10.1145/333329.333344`.

[11] Joseph J. LaViola. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32 (1 Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: `10.1145/333329.333344`.

[12] Young Youn Kim et al. “Characteristic changes in the physiological components of cybersickness”. In: *Psychophysiology* 0 (0 Aug. 2005), 050826083901001–??? ISSN: 0048-5772. DOI: `10.1111/j.1469-8986.2005.00349.x`.

[13] Clare Regan. “An investigation into nausea and other side-effects of head-coupled immersive virtual reality”. In: *Virtual Reality* 1 (1 June 1995), pp. 17–31. ISSN: 1359-4338. DOI: `10.1007/BF02009710`.

posure to redirected walking in virtual reality.[14] There will most likely be a overlap in the factors, however certain factors might have a stronger correlation to negative symptoms in redirected walking than just regular virtual reality experiences.

The important question is why the users suffer/don't suffer from this and also how we minimize the risk of such incidents occurring. Factors such as redirection algorithm could be an interesting place to look at this. Some redirected walking algorithms, such as the steering-based algorithms that steer the user by rotating the view, might be harsher on the user compared to redirection using audio based redirection with the help of stereo sound, which could be easier on the user, perhaps blinking might alleviate the cybersickness symptoms in the same way how a virtual nose helps motion sickness in virtual reality[15].

2.9 Understanding "VR-Sickness"

The concept of "VR-Sickness" is generally thought as to be a umbrella like set of unpleasant effects which users experience from using a VR headset. This can occur both during the use of VR and after the use of VR. Either directly after finishing the use of the VR headset, or a short time frame after. One of the goals of this thesis is to identify what symptoms VR-sickness is composed of, and to understand if the technical choices in the redirection algorithm has any impact on both the prevalence of these symptoms, and their intensity.

First, in order to perform an in-depth discussion of this phenomenon, it is important to review earlier literature describing motion-sickness in simulators.

---

[14] Jann Philipp Freiwald et al. "The cybersickness susceptibility questionnaire". In: ACM, Sept. 2020, pp. 115–118. ISBN: 9781450375405. DOI: `10.1145/3404983.3410022`.

[15] Carolin Wienrich et al. "A Virtual Nose as a Rest-Frame - The Impact on Simulator Sickness and Game Experience". In: IEEE, Sept. 2018, pp. 1–8. ISBN: 978-1-5386-7123-8. DOI: `10.1109/VS-Games.2018.8493408`.

As before the proliferation of VR headsets came the simulators. Simulators are large mechanical constructs designed to simulate how it is to drive a car, commercial aircraft or fighter-jet. They typically feature inclination control, realistic field of view, speakers and haptic feedback. The big difference is that the user sitting down, and not wearing a VR-headset.

Sickness that originates from the use of simulators were typically referenced to as simulator sickness. Each simulator might have different parameters, such as maximum and minimum tilt of the simulator platform, the intensity of the haptic feedback and field of view provided. Simulators designed for cab-driving are not the same as simulators for commercial aircraft, or for fighter jets. As such, it is important to identify common symptoms which occur in all instances that are not necessarily due to the simulator specifics.

Early research on simulators come from research done on vertical simulators. These are designed to expose the user to g-forces in the vertical axis, common during combat maneuvers in military aircraft or simply from turbulence.[16] Such research found that a plethora of symptoms were reported. In the case of one more modern paper, they identified 25-30 factors which contribute to simulator-sickness.[17]

The advance from these vertical simulators, to aircraft simulators and eventually virtual reality, which can be considered a form of simulator. The sit-down VR experience can represent a simulation. If the user is exposed to a roller-coaster, the anticipation of taking a turn will make the user brace. In the case of stand-up VR, the same applies. Taking a step over a cliff side

---

[16] S. J. Alexander et al. "Studies of motion sickness: XVI. The effects upon sickness rates of waves of various frequencies but identical acceleration." In: *Journal of Experimental Psychology* 37.5 (1947), pp. 440–448. DOI: `10.1037/h0063240`. URL: `https://doi.org/10.1037/h0063240`.

[17] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

creates the same response, so the use of ”simulator” sickness questionnaires is a valid approach, and already used with certain modifications.

2.10 Simulator Sickness Questionnaire

Common symptoms/factors of VR-sickness typically include vertigo, nausea, disorientation and headache. These symptoms are common among VR-sickness research where the user is subject to some experiment in sit-down VR experiences. The user is exposed to some VR-experience, and they asked are to rate how affected they are by these conditions (see section 3 either during or after the experiment. The intensity is also reported by the user, how the intensity is determined is usually self reported.[18]

The SSQ is a system developed for assessing the usability of military aircraft simulators. The SSQ is a table which has the diagnosis on the Y-axis and the intensity expressed on the X axis. The intensity is enumerated typically as ”Low, Medium and High” degrees of severity. Some instances opt for a ”No Feeling” option, which simply means that the user did not feel that particular symptom. However the absence of an answer on the particular row implies the same as such an option. Note that how the SSQ is applied will be described in the user study design. For now, the SSQ is used as a reference for discussion for what VR sickness is.

Symptom	Mild	Medium	Heavy
General Discomfort	0	X	0
Fatigue	X	0	0
Vertigo	0	X	0

Table 1: Standard SSQ form with only three symptoms

---

[18] Hyun K. Kim et al. “Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment”. In: *Applied Ergonomics* 69 (2018), pp. 66–73. ISSN: 0003-6870. DOI: `https://doi.org/10.1016/j.apergo.2017.12.016`. URL: `https://www.sciencedirect.com/science/article/pii/S000368701730282X`.

According to the original paper by Kennedy, the SSQ is supposed to be administered post-test.[19] Many studies that use SSQ disagree with this approach, and choose to administer it pre-test and post-test in order to establish an baseline.[20][21]. The pre-test and post-test form is the same, and is shown in figure 3. For this thesis, it is administered post-test for each scale and/or algorithm, then compiled so that simulator sickness questionnaire data is available for any combination of scale/algorithm.

The SSQ scoring system consists of three sections. The sections nausea, oculomotor and disorientation. The symptoms which are asked in 3 are divided into these sections based on how close they relate to the sections. Nausea would end up in nausea alongside with dizziness, while degraded balance and AVS would end up in disorientation. Where each symptom is placed is described in[22]. Each of these sections represent a sub-sum in the SSQ formula. The scores for all three sections are multiplied by a weighting value, which is different. In the original Kennedy paper, the weights are set so that symptoms in the disorienting category weigh move than the lower sections. The lowest weight is the oculomotor section, while the highest is

---

[19] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

[20] Pauline Bimberg, Tim Weissker, and Alexander Kulik. "On the Usage of the Simulator Sickness Questionnaire for Virtual Reality Research". In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2020, pp. 464–467. DOI: `10.1109/VRW50115.2020.00098`.

[21] Stéphane Bouchard, Maxine Berthiaume, et al. "Arguing in Favor of Revising the Simulator Sickness Questionnaire Factor Structure When Assessing Side Effects Induced by Immersions in Virtual Reality". In: *Frontiers in Psychiatry* 12 (Nov. 2021). DOI: `10.3389/fpsyt.2021.739742`. URL: `https://doi.org/10.3389/fpsyt.2021.739742`.

[22] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

the disorienting section. The idea is that for Kennedy's simulators, which were aircraft and helicopter simulators, having trouble focusing vision is not as important as becoming nauseous or disoriented, therefore the different weights.

The formula from Kennedy's paper is as follows, where n, d and o represent nausea, disorientation and oculomotor respectively:

$$Score = (n * weight) + (d * weight) + (o * weight)$$

The weights are predefined from aeronautical psychology paper which performed a large scale analysis on several US Navy flight simulators[23]. For thesis, the weights are not important, as it is possible to still do the perform the SSQ without weights[24], as the original SSQ procedure by Kennedy et al with weightings on factors based itself on comparing symptoms between trained personnel, meanwhile the alternative SSQ by Bouchard et al relates to comparison between untrained personnel. The term "trained" or "untrained" personnel in its original sense relates to civilian or military aircraft personnel. In thesis, we make no such distinction, therefore the use of Kennedy's model would not matter. Regardless, papers using SSQ list a series of symptoms which serve as a good starting point.

The second part of the SSQ procedure is represented with the figure 2.10. This part is not used for thesis, due to the fact that the people who are going to participate are not guaranteed to be people who have experience with the

---

[23] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

[24] Stéphane Bouchard, Maxine Berthiaume, et al. "Arguing in Favor of Revising the Simulator Sickness Questionnaire Factor Structure When Assessing Side Effects Induced by Immersions in Virtual Reality". In: *Frontiers in Psychiatry* 12 (Nov. 2021). DOI: `10.3389/fpsyt.2021.739742`. URL: `https://doi.org/10.3389/fpsyt.2021.739742`.

simulator technology in contrast to the original Kennedy SSQ, so a similar weighting system cannot be calculated and used.

Symptom	Nausea	Oculomotor	Disorentation
General Discomfort	1	0	1
Fatigue	0	0	0
Vertigo	1	1	0

Table 2: Standard excerpt from SSQ with select symptoms

The symptom that an SSQ typically consists of is as follows. This example is from the original simulator sickness questionnaire paper[25]

- General Discomfort†
- Fatigue
- Eye strain†
- Difficulty focusing
- Sweating†
- Nausea
- Difficulty concentrating
- *Fullness of head†*
- **Dizzyness (Open-eyed)**
- **Dizzyness (Closed-eyed)**
- **Vertigo**
- Blurred Vision
- **Stomach Awareness**
- **Burping**

and the minimized SSQ which is used for the study

---

[25] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

- Nausea
- Migraine
- Benign paroxymal positional vertigo (BPPV) (I feel like spinning)
- Vertigo
- Worsened sense of position (Acute Vestibular Syndrome) (I don't know where I am when I close my eyes)
- Emesis
- Non-critical loss of balance
- Critical loss of balance

The list of symptoms above is typically what a SSQ consists of. For studies where the focus is on "Cyber-sickness/VR-Sickness", the list is repeated. However the list items consist of symptoms where they may be compressed, or represented as a different symptom. For this thesis, the proposed series of symptoms is as follows:

In the original list, the items in bold text have been combined with adjacent symptoms. Items with a dagger mark were removed, either due to being too generalized, difficult to observe in the experiment participant or expected as a common effect of having a VR-headset close to the participants face.

From the initial list, the following symptoms were removed: General Discomfort, Eye strain, Sweating and fullness of head. The reasoning is as follows: For "General Discomfort" the symptom is too broad, and is already covered as one of the points in the SSQ variant used for this thesis. In particular, discomfort is a Likert scale from 1 to 5 where 1 indicates no discomfort/feeling of sickness and 5 indicates sickness to the point of aborting the test in order to preserve the tester's well being. This preserves the granularity of discomfort, and makes it easier for the observing personnel to collect this measurement.

Nausea is a very general term, however it is important to note this particular symptom, as it appears as a precursor to many of the more complex cases, such as BPPV, AVS and so on. Nausea can be used as an indicator that some technical variables, such as the type of redirection and gain can have effects. The other cases, such as BPPV, AVS and similar may occur, but most likely result in a lower prevalence rate than simple nausea. So its use as an standalone metric is important, both for statistical methods (for the SSQ) and as an indicator.

The categories "dizziness (open-eyed)" and dizziness (closed-eyed) have been combined into benign paroxysmal positional vertigo (BPPV). The reason behind this, is to reduce the amount of symptoms by merging them into one category. In addition, BPPV encapsulates the condition better than just the two symptoms. The same rationale is applied to produce "Worsened sense of position", known as "Acute vestibular syndrome" in medical terms. The two describe the same phenomena, however the difference is that BPPV does not manifest itself when moving. Say, during the VR-testing, the user feels they have a spinning/dizzy-like sensation in them. They move and during locomotion, this feeling subsides. This is how BPPV is described in the medical literature[26]

AVS on the other hand will be something the user feels, even when standing still. Another critical, but differentiating characteristic is that AVS lasts much longer then BPPV, typically one of the characteristics that the feeling lasts for around 24 hours or more. AVS is also referred to as "vestibular neuritis". It is important to consider both cases, as the VR-tests feature unrestricted movement, instead of the classic sit-down VR experience. If testing was restricted to sit-down VR experiences, then it would be sufficient only to test for AVS and not BPPV.

During this study, observing the user for more than 15 minutes after test conclusion is difficult in order to be able to differentiate between BPPV and

---

[26] Peter Johns and James Quinn. "Clinical diagnosis of benign paroxysmal positional vertigo and vestibular neuritis". In: *Canadian Medical Association Journal* 192.8 (Feb. 2020), E182–E186. DOI: `10.1503/cmaj.190334`. URL: `https://doi.org/10.1503/cmaj.190334`.

AVS. Normally, differentiating between these two cases, it is necessary to perform a field test. This test is called the Dix-Hallpike test. Due to the lack of certified medical personnel, and to avoid uncomfortable tests; as the Dix-Hallpike itself can be slightly violent, the test is omitted. Instead, cases of vertigo where the vertigo manifests during movement are to be registered as AVS, as BPPV is short term, typically lasting only 20 seconds or less.[27]. Before the tests, the user is informed that they can quit the test at any time, for any reason, and it is emphasized that they should quit if they feel unwell to the point where it impacts their well-being.

Critical and non-critical loss of balance is defined as follows. Non-critical balance is defined as an incident where the test-taker momentarily loses their balance, and must shift their weight, either by rotating themselves, throwing their arms out to act as a counter-force, or otherwise compensate. Critical loss of balance is defined as a case, where the test taker falls on the ground due to the loss of balance being so severe it is not possible to correct by shifting position or using the arms as a counter-force. In critical cases, it is important to aid the test taker in the case of injury. If no injury is observed after rendering aid, the test-taker is asked if they wish to proceed with the test by skipping the current VR-level. They are also allowed to quit the test. In non-critical cases, the test-personnel will verbally ask if the test-user is fine, and also present the option to terminate the test, skip or proceed as usual. The test personnel will attempt to prevent critical loss of balance by following the user, but avoiding them as to not be a trip-hazard. In addition, the testing area contains several safety features, which will be described later.

Feeling of emesis / emesis is as simple as verbally asking the user if they feel sick, and if an positive (as in sick) answer is retrieved, then the user is asked

---

[27] Jonathan A. Edlow, Kiersten L. Gurley, and David E. Newman-Toker. “A New Diagnostic Approach to the Adult Patient with Acute Dizziness”. In: *The Journal of Emergency Medicine* 54.4 (Apr. 2018), pp. 469–483. DOI: `10.1016/j.jemermed.2017.12.024`. URL: `https://doi.org/10.1016/j.jemermed.2017.12.024`.

to describe what they are feeling. If the answer is ”I feel like vomiting”, then the user is asked if they wish to abort the test, skip the current VR-level or to proceed. If the user does throw up, then the test must be aborted, as the test-taker is too ill to proceed. In addition, this means that the test area is unsuitable for further testing, until the area is cleaned.

In order to differentiate, the following field tests have either been found in the medical literature or made in order to differentiate the severity. They are quick to perform, require low to no medical knowledge and no physical contact. They are as an aid for the test observer to score the symptom to a three point grade.

- **Symptom: Fatigue**
  - **Mild** The user has started noticeably slouching when they have previously not
  - **Medium** The user mentions/complains about neck or headset weight in non-passing way
  - **Heavy** The user is no longer able to support the weight of the HMD
- **Symptom: Nausea**
  - **Mild** A very light feeling of unease is reported by the user
  - **Medium** A stronger feeling of nausea is reported, and hinders task performance
  - **Heavy** THe nausea effect is so strong, that the feeling of imminent emesis is reported and/or test is requested to be aborted.
- **Symptom: Migraine**[28]
  - **Mild** The user has feels the oncoming of a headache or a light headache

---

[28] Ottar Sjaastad et al. “Grading of headache intensity. A proposal”. In: *The Journal of Headache and Pain* 3.3 (Dec. 2002), pp. 117–127. DOI: `10.1007/s101940200029`. URL: `https://doi.org/10.1007/s101940200029`.

  - **Medium** The user has a noticeable headache, and is impacting their focus on the test
  - **Heavy** The headache is so strong, that the user does not wish to continue

- **Symptom: Benign paroxysmal positional vertigo**
  - **Mild** The user has feels a slight ”spinning” sensation
  - **Medium** The user feels a strong spinning sensation, even with eyes closed.
  - **Heavy** The spinning sensation is so strong, the test is stopped or skipped.

- **Symptom: Degraded Balance**
  - **Mild** The user reports that they feel a bit unsteady on their feet, or a mild ”wobbly” feeling
  - **Medium** The user reports unsteady footing, and can explain how this feels
  - **Heavy** The user is wobbling significantly, and can be observed by test personell

- **Symptom: Emesis**
  - **Mild** The user reports a feeling of nausea accompanied by a feeling of wanting to throw up
  - **Medium** The user mentions that they are going to throw up.
  - **Heavy** The user has thrown up.

- **Symptom: Non-critical loss of balance**
  - **Mild** Test personnel observe that the user trips slightly, but without falling.
  - **Medium** The user trips and has to balance themselves using their arms.
  - **Heavy** The user trips, and almost falls, corrects by balancing using their arms

- **Symptom: Critical loss of balance**
  - **Mild** Test personnel observe that the user trips and lands on their knees or arms
  - **Medium** The user trips and is unable to land on either knee or hands
  - **Heavy** The user trips in a flailing manner and collides with the ground.

The symptoms mentioned earlier in section 2.10 is typically what "VR-sickness" is composed of[29]. It has a significant overlap with simulator sickness. However another question arises, and that is *how* VR-sickness happens. Both from a pathological perspective, and from a technical perspective. The pathological perspective is best reserved for medical professionals who are able to perform more in depth tests on the human body to understand the effects. The technical perspective is something that can hopefully be answered without medical tests, and with statistical tests.

3 Experiment Setup

Testing the effects and to understand what technical factors contribute to the incidence rate of unpleasant effects, such as cyber-sickness, an testing environment must be established. In this thesis, we have chosen Unreal Engine as our testing environment/engine. Common uses of Virtual Reality is in entertainment and multimedia applications, such as video games. Unreal Engine is commonly used to create such applications, and as such the Unreal Engine may be combined with the redirection algorithms to create a gamified approach where the user plays a video game which involves physical locomotion in a relatively open space.

---

[29] Hyun K. Kim et al. "Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment". In: *Applied Ergonomics* 69 (2018), pp. 66–73. ISSN: 0003-6870. DOI: `https://doi.org/10.1016/j.apergo.2017.12.016`. URL: `https://www.sciencedirect.com/science/article/pii/S000368701730282X`.

3.1 Unreal Engine

Unreal Engine is a popular game engine developed by Epic Games. The engine provides us with a way to interact with the VR headsets and comes with an generalized API (Application programming interface) to these headsets. Programming in Unreal consists of writing C++ source files which the engine bundles and compiles to an executable format suitable for the target environment. In addition to the ability to program in C++, there exists a drag and drop programming system called "Blueprints" which are explained as an quick and easy way to create game-play scripting. For our approach we have combined both C++ code and blueprints in order to speed up the creation of game-play logic while C++ has been used for logging of position and to some extent, for redirection systems themselves.

3.2 Choice of VR headset

The choice of headset is important, as each headset provides different strengths and downsides due to the type and amount of hardware sensors, ease of deployment, processing power and the capacity of the built in power source. The feature-set provided by newer headsets is useful for better redirection. Some examples are compatibility with beacons worn on the body, which lets the headset track the direction of body parts independently of the headset, or optical tracking of the fingers, allowing the user to not use a controller, provided that the hands are visible to the optics. Such features provide interesting and use full approaches to redirected walking, however come at a downside of increasing the cost of the headsets.

There has been a split between ease of deployment and processing power. One company, for example Meta, seems as if they have angled their VR headsets towards a less technically competent market. The rationale behind this claim comes from the tight integration with their Meta account platform, high portability and less reliance on external sensors. One significant advantage is the "Air-link" system which the Meta quests can use. While normal operation is restricted to cabled operation, it is possible for the headset to communicate with a computer in the same LAN over 802.11 technology. This allows wireless operation without the use of heavy first-party adapter/battery bundles. The headset and computer need to be in the same local network, and the headset must have (at the time) experimental features enabled. The user simply selects "Air-link" in the connectivity section of the headset, and confirms the pairing at the computer which is to be linked. Afterwards the headset and computer is paired and applications may be "streamed" to the headset.

There is however a few caveats with this Air-link system. First, the use of a 5GHz network is highly recommended by the headset. A notification is provided to the user that the use of a 5GHz band is optimal, and the use of other bands may lead to worsened user experience. The second issue is that this system uses more of the headsets battery pack, and therefore less uptime can be expected. Lastly, the computer linked with the headset must have a CPU strong enough to both handle Unreal and the actual handling of data to be sent to the headset in reasonable time. Not all computers are quick enough to satisfy this requirement, and as such this means a heavy, powerful computer must be available for use in this context.

Another company such as HTC aims towards the more technically competent market, but with less portability. One such example is the HTC Vive Pro series of headsets, which requires sensor beacons in order for the headset to operate, but comes with stereo headphones built into the headset. It also features wireless features with an additional upgrade package which consists of a battery-pack and wireless adapter which needs to be mounted above the head-strap, and an adapter to be installed into a PCIe slot. This system operates over millimetre-wave technology. This headset competes in the power-user market, which may have computers available with stronger discrete graphics processors and spare PCIe slots if the wireless option is to be employed.

The last headset to be considered but sadly impossible to get in Norway is the Valve Index. The Index is a VR headset developed by Valve Corporation. While the headset itself does not have any unique features compared to the Vive or Meta Quest, it has an interesting input device, the "Index Knuckles".

The Knuckles come with individual digit tracking and haptic feedback, which allows the system to track the position of each individual digit. The device itself is secured to the hand, specifically using a band that loops around the hand slightly behind the knuckles, hence the name. The hand then wraps itself around the controller joystick and allows the use of the controller, which consists of a joystick and two buttons, but also allowing much more immersive controller use. Unfortunately, Valve's products, for an unknown reason, do not ship to Norway, and acquisition of their products would require a PO box in Sweden. So the possibility of more immersive input from the Index Knuckles is not an option.

The choice of headset may be summarized into a table as follows.

Headset Name	Connectivity Type	Controller Type	Tracking Type
HTC Vive Pro	Cable or first party wireless.	Two joysticks, tracked by headset	Tracking Beacons
Oculus Meta	Cable or wireless	Two joysticks, tracked by headset	SLAM
Valve Index	Cable or aftermarket wireless	Two joysticks, tracked by headset	Tracking beacons

While the option of using the HTC Vive Pro alongside the Index "Knuckles" controller seems like a great choice to maximize immersion, it is sadly not the most optimal solution for our testing environment. As there is the logistical issue of ordering the knuckles to Norway, and the fact that a headset with a cable that is not detachable (or at the very least without taking apart the headset assembly) means that a cable extension through a boom arm is needed to avoid the cable tugging or wrapping around the user during use.

This means that headsets with wireless capability is the only viable choice for our experiment setup.

The Sustainable Immersive Networking Lab (SINLAB) where we develop our redirection algorithms has kindly allowed us to use their VR headsets. The devices available to us consist of the HTC Vive Pro, alongside the tracking beacons for the Vive Pro and a Meta Quest. (Formerly Oculus Quest before the acquisition by Meta) Early on during the development cycle, the Vive Pro beacons malfunctioned and stopped emitting infra-red light, which was used to track the Vive pro. The reason for this is unknown, and has removed the Vive pro from the pool of headsets that could be used.

For the headsets, which seem to run a modified Android operating system, Unreal produces an .apk file which is automatically installed onto the headset, and ran as an standalone application. In this way, the headset can contain the compiled project as an Android application allowing us to run user studies without having the user feel or perceive either the tugging of a cable, or to come in contact with a cable by having it brush against their legs or hands. It will also allow us faster and more efficient testing, by making the tests not need to be reset. The personnel deploying the experiment can simply close the program, and start it again. This gives us a slight improvement when it comes to immersion, and simplifies deployment of the experiment. If the headset was required to be cabled, then an long cable alongside with a boom-arm might have been necessary to avoid tugging and contact with the cable

In addition to the compilation, it is possible to run the project directly in the headset, albeit at an increased computing cost, as the optimizing benefits of compiling are not utilized. This simplifies the development process, as we do not have to wait for compilation, transfer and installation in the Android operating system which the headsets run. This is beneficial because it makes development and testing much more rapid, as compilation of the project may take several minutes compared to testing of a feature which may take a fraction of said time. A figure describing the two approaches can be seen at figure 1

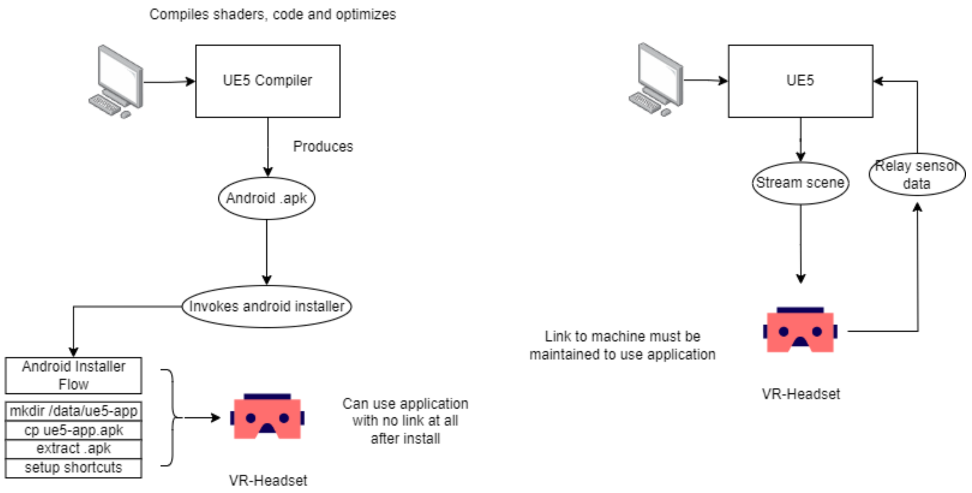


Figure 1: The compile and install process (left) versus running the project directly in the headset over a stream (right)

3.3 Map design

As mentioned earlier, the approach that is chosen to study the effects of virtual reality redirection and its pathological effects on users is to gamify the experiment. The test takers will receive a series of tasks, and will be required to complete them to the best of their ability. The reasoning behind this is that many use cases for virtual reality and redirected walking relate to entertainment purposes, and it is wise to observe the prevalence of such negative effects in an entertainment context. Basic design of these ”maps”; environments in which users can explore and interact with the world will be described and why the map is designed in the way it is. In addition to this, a basic taxonomy of maps will be described.

In the Unreal Engine, map design is a central component. Maps are represented as a collection of meshes and textures. The game designer creates a map by placing meshes; which are objects with a pre-determined shape and applies textures to said meshes to make the mesh look like some scene object. Unreal comes with a set of basic geometric mesh shapes, consisting of both

3D and 2D instances. Some 3D meshes are cubes and spheres, and some 2D meshes are planes and circles. While the choice may confine the designer into making block-like and otherwise boring structures, unreal ships with a powerful set of tools under the ”modeling” section. Modeling tools allow the map creator to take these basic meshes, and create new ones by combining two meshes together, or using one mesh to hollow out another mesh. These operations are called mesh booleans, and are fundamental in many modeling tools. The designer can also extrude meshes. Extrusion of a mesh is done by first selecting a base mesh, then drawing the shape of the new mesh on top of the base mesh. This can be done to create cubical, spherical, polygonal or freehand meshes. After the extrusion is completed, the mesh can then be pulled out of the base mesh to create a extrude it out, or pushed in to hollow out the base mesh. The design of say, furniture can be created in a very easy way using this method. A sofa may be created by taking a rectangular mesh, and applying a Boolean with a smaller cubical mesh. In the case of figure 6; the cauldron, it is created by taking a two spheres, one larger and one smaller. The smaller sphere is then inserted into the larger sphere, and the boolean operation is specified so that the new mesh is the space occupied by mesh A minus mesh B. Unreal engine then creates the new mesh and discards the sacrificial mesh B by default.

While the use of mesh modeling tools gives significant freedom, it is at a performance cost. The cost of combining meshes is that the vertices that the mesh consists of is added for each mesh merge operation. If the sofa mentioned earlier consists of 32 meshes, then the mesh count after the boolean will increase, as the comparably more complex shape, despite having less area, will contain more vertices to describe its shape. High vertex counts lead to worsened performance, and must be used in a way that minimizes the amount of vertices, or in a way that creates minimal increases in vertex count.

This is important in order to minimize the computation cost of rendering

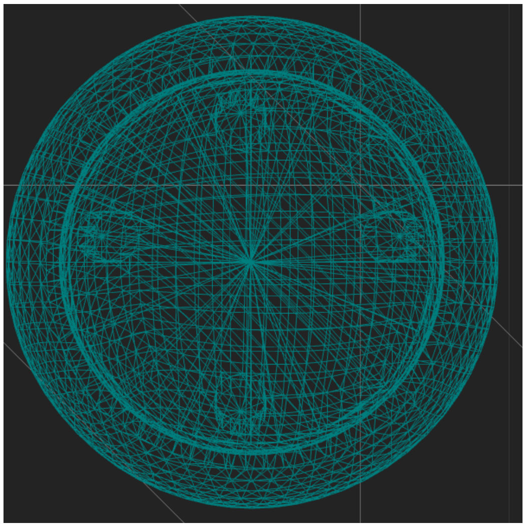


Figure 2: The vertex view of the cauldron and its legs. Note the relatively very dense vertex-mesh. It consists of 3600 vertices

such meshes on the VR headsets. This can be accomplished by the polygonal group tools. The polygonal group tools allows manipulation of individual or groups of vertices. The user selects a set of vertices, then they can subdivide them (this splits the vertex into two) and manipulate the newly created point by pushing or pulling said point.

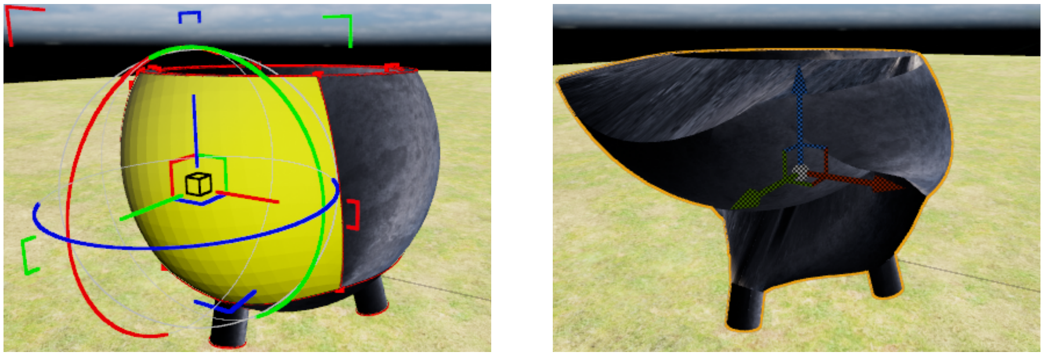


Figure 4: Rotating the selection to warp the mesh.

This allows the user easily create disfigured meshes, or if applied correctly to curve and warp meshes without increasing vertices. It should be said however that vertex counts can be quite high, but anecdotal experience says that vertex counts above 20 thousand vertices will start to be too much. Such a count represents detailed models that are to be placed on players, while 2000 to 6000 are for environmental meshes, such as the cauldron. Regardless, the use of polygonal group tools is best used to slightly alter meshes, not create completely new ones. It is a time expensive process, which produces great results, but it is all up to the designers choice.

When it comes to design of the terrain, then the previously mentioned tools may be used. This will only create square or warped terrain, as the use of polygonal tools and booleans is both waste of time when landscape and terrain is to be made. Creating a slope is difficult, as it will warp and be uneven. Unreal has a solution to this under the "landscape" toolkit.

Landscaping tools work by placing a special set of meshes on the ground. These special meshes are called landscape proxies. A landscape proxy is a collection of meshes, grouped together in what is called a quad. Quads then form sections. These quads are well optimized cubical meshes which can be manipulated using similar techniques that polygonal tools apply to standard meshes. These tools are designed to emulate how natural processes shape terrain in real life. The designer can sculpt, apply different erosive models, such as rain or wind, smooth en terrain or elevate terrain to create mountain-like terrain. It is also possible to apply height-maps to terrain to automatically copy real life terrain elevation into the map.

3.4 Taxonomy of map and task design

Maps are a crucial component of any game. They represent the environment in which users can explore and interact with the game-play aspects. A map without game-play aspects is simply just an open space in which the user can navigate. Such instances are not very engaging to the user, as there is

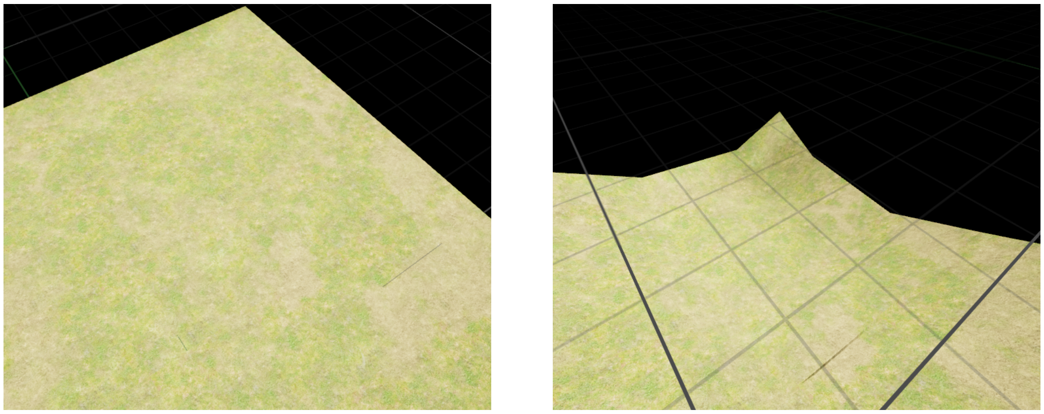


Figure 6: Rain based erosion on the proxy

no entity trying to keep the user engaged and immersed in the map. While it can generally be said that only having a map and no game-play aspects is ”boring” or not ”engaging”, it is important to consider where the this entity in which keeps the user interested arises from. Suppose there are two maps, a gigantic empty grass plain and a abandoned house. The gigantic plain may be interesting to be, as it is open and vast however there is nothing to explore. In this scenario the ”entity” arises from the user, curiosity, which may be short lived as there is nothing to explore. The house however, even though it is abandoned, could still be engaging to the user to explore. Do they know it is abandoned, are they expecting something to happen, is there something in the attic or basement? Like the plains, this entity arises from the user and is manifests as curiosity. The user is curious and explores the map.

This describes one branch of this taxonomy. Maps which rely on the user to self-engage are typically described as exploration maps. It is important to realize that a map can be a combination of several taxonomic aspects. They are not mutually exclusive, a map which is inviting to the curious mind may also combine itself with other aspects. Another aspect is a task based aspect.

Task based aspects are aspects in which there is a task to be performed in the map, therefore the user must try to complete it. Why they might do so could be because they wish to advance the level, to obtain some special item or simply to see what would happen. Practical examples of task based aspects could be bring an item to a specific position, move to a trigger or to solve a puzzle. Task based aspects are very common in video games. It could be said that a video game is solely composed of task based aspects, and in the case of the experiment, it is also composed of completing tasks. The idea is to let the user explore or solve the tasks in ways which they deem to be the most efficient, and their condition while doing so.

The experiment consists of the user playing through several task based maps, in which the intensity of the redirection algorithms varies in a random fashion from map to map. The order of maps is not random, as the user starts off in an easy, non-cluttered environment so that they may familiarize themselves with how to move in VR and how to interact with objects. The subsequent tests feature more complex environments, in which the user has to move between several randomly assigned points to accomplish the task. After the user has finished the task, they have to complete it again with a new random set of points to visit and changed intensity of redirection.

3.5 Design of maps

The first test (for the pilot study), named the baseline test, is a test in which the user is told to follow a straight path. After finishing the path, they are prompted in the virtual world to describe if they either went in a straight line, a line that curves to the left, or a line that curves to the right. The point of the baseline test, which is a form of task based test, is to see how well the users tolerate redirection in a simple and controlled environment. Such baseline tests may help determine if the users are bad candidates for virtual reality if they get sick. They are regardless allowed to proceed in the test as long as they feel comfortable enough. In addition to acting as a filter, it also allows observing how inconspicuous the redirection is. If the user does not notice the redirection, completes the test, and chooses "I went in a straight line", then this may be used to support the claim that the user was immersed, and did not notice the redirection.

The entire point of the baseline test is observe the user in a situation with a minimal amount of distracting elements and simple tasks to be completed. The choice of redirection algorithm in this case can either be an instance of a steering algorithm, such as steer to center or steer to orbit or a translation based algorithm such as translation (also known as one-to-N mapping). The algorithm itself is not all that important unless one wishes to prove its efficacy by only using the baseline test itself. For this thesis, the goal is to identify what technical factors, such as algorithm and/or intensity of algorithm cause negative effects. The baseline test allows us to test for these factors in an controller environment.

The first test (the first actual test used in the main study) is called "the zigzag test". The zigzag test is a test where a series of buttons are placed in an zigzag pattern in a hallway. The test taker has to navigate to the buttons, in order, and press them. After the buttons have all been pressed, then the user can proceed to the next test. The principle here, which is a task based test, is to see how well the user does in a slightly complex environment. The environment is designed to slightly aid the redirection algorithm by having the user follow a zigzag pattern. In the case of steer to center, the user is regardless going to be turning to face the next button, which gives an opportunity to apply the redirection as they are going to be turning. The map design attempts to directly aid the redirection, by making sure the tasks are laid out in a way so that its highly probable that the user will turn sharply to the next button. This will might allow higher gains to be used, as the predictable path complements the higher gain. The ziz-zag test is partly inspired by the ”Fire-drill” test, by Sharif Razzaque.[30]

The second test is called the track and field test. This test places the user into a large running track with a goal line placed in the distance. The user is supposed to walk towards the goal line in order to finish the level. Note that even though this is a running track, the user is instructed to not run, as it is difficult for the user to react, stop and turn if they exit the boundaries. Most likely, the user will not be able to decelerate in time to avoid colliding with a wall in the physical space. In addition, it makes it difficult for the test personnel to stop the user, even when following the user. The test can be used to test rotational, curvature and translation algorithms. The test is also easy and intuitive for the user.

The third test, dubbed ”The witch cauldron” is a test where the translation algorithm is tested. Translation, as mentioned earlier is where a displacement in the real world is reflected as twice the displacement in the virtual world. When it comes to the contents of the experiment, it is as follows. The user is given a order of ”orbs” to collect. In the actual game world, the order is a set of colors. The user is prompted to bring the red orb, place it in the cauldron and wait a few seconds for the cauldron to dissolve the sphere. If this is the correct sphere, the next orb to be collected is revealed. The user is shown which orb to bring by having a floating, spinning orb above the cauldron. If the user places the wrong orb in the cauldron, then a explosion sound effect is played, and the orb is launched out of the cauldron. The orb will roll away, or bounce, depending on how the game engine simulates the physics and the user must then collect the correct orb before placing the orb that was wrongfully placed back in the cauldron.

Once all four orbs have been placed, the level restarts. The user can also chose to ignore the orbs, and look around in the house. Several ”distraction”

---

[30] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. *Redirected Walking*. 2001. DOI: `10.2312/EGS.20011036`. URL: `https://diglib.eg.org/handle/10.2312/egs20011036`.

objects have been placed, and windows have been placed which give a view towards green mountaintops. The idea behind this is to see if the user can still avoid walls in the physical world by not following the predicted paths, such as the shortest one from their position, to the current orb to be collected.

All three maps are illustrated in the figure below. Note that the order in which the maps appear is randomized and do not relate to how the images are shown.

Figure 9: The three maps. Left-Right: Zig-zag, Track and field, Witch Cauldron

One important aspect shared among all of the maps (aside from the track and field) is that the map design (or rather the task based aspect) complements the redirected walking algorithm. Take for example the third test; the witch cauldron. While the house allows the user to navigate freely inside, the actual task itself creates a set of predictable patterns similar to a graph. Going to one of the tables and picking the correct color to place in the cauldron means that the user will go to the cauldron, and return. Then the user can pick three paths, the paths that are not directly opposite can form a curved path, like figure 3 illustrates, and the paths that cross the cauldron in the virtual world, will produce curved paths due to the user trying to avoid solid objects as if they were actual objects in the physical world. Therefore any path that pertains to the task will produce a path that can be curved, and thus aid the redirection algorithm. The only exception to this case is if the user decides to explore the house and look outside the windows, as the paths produced

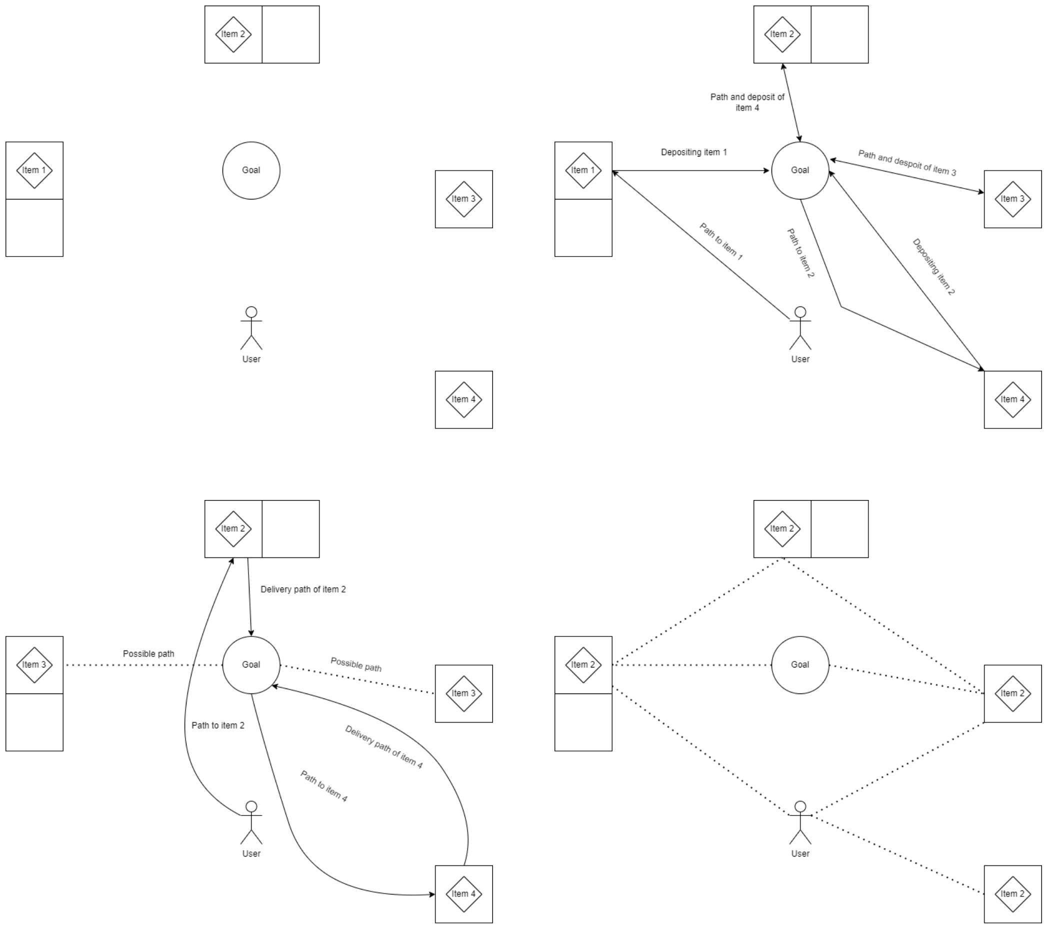


Figure 10: Generalized setup and possible virtual paths to accomplish task

from a window to a table; where the spheres are placed, do not have such guarantees. The first test also exhibits these properties, due to the zigzagging placement of the buttons. The exception to this case is the track and field, however a case could be made that the user focusing on getting closer and closer could distract them, so that they do not notice rotation or curvature algorithms being applied to their movements.

3.6 Design of user study

The study is designed in a two pronged fashion. In the experiment, objective data is being collected from the headset sensors, and is stored on the computer hosting the Unreal Engine instance. Subjective data is collected by asking the test taker after each VR-level a set of questions relating to their experience with the system (Do you feel as if the system is something you could use for other games?), a question relating to how unwell they feel (From a scale from 0 to 5, where 0 is not unwell, and 5 is very unwell? how unwell do you feel) and if the answer is greater than 0 on the sickness prompt, they are asked to describe what they are feeling. The prompt for for describing does not ask "sick", as to not make the user think they are supposed to feel sick, and possibly misinterpret something as sickness.

The person asking the question is supposed to match the closest symptom that the user is feeling to the modified SSQ (Simulator-sickness questionnaire). Attached to the form used by the experiment personnel is an prompt which the personnel is supposed to ask. An example to such an prompt, for AVS (Acute vestibular syndrome) is "Do you feel like something is pulling you when walking?" instead of describing the formal criterion for the symptom. This allows the test takers to not focus on what medical characteristics the symptoms are composed of, and instead parse the answer that the person undertaking the test returns.

It is important to note that the user is not given a battery of prompts immediately after each test. They are first asked the following prompt: ”Do you feel a sense of discomfort”. This is to assert if the user feels at unease without asking the user if they feel sick directly. The idea is to avoid asking if they are directly sick, as the user might think that they are supposed to be sick, and might interpret any unusual stimuli as sickness.

Not all symptoms in the modified SSQ has such a prompt, some are visual queues that the person administrating the test needs to observe. In the case of critical and non-critical loss of balance, then the person administrating the test needs to simply observe, and in some cases ask and verify if the loss of balance was not due to external factors, such as a slippery floor, untied shoelaces or other reasons not relating to the VR experience.

When an answer is returned, the person administrating the test also needs to discern how critical the symptom is. The SSQ provides a matrix, where the type of symptom is on the x axis, and the severity is on the y axis. In order to illustrate, assume that the person undertaking the test answers that they feel nauseous. The person administrating the test now needs to rate the criticality of this nausea from three degrees of seriousness; mild, medium and heavy nausea.

In order to differentiate between the three degrees of severity in the simulator sickness questionnaire, simple field tests can be used by the test personnel to assess the functionality of the senses, or to see how worsened they have become; the dysfunction of the senses. This is to be applied when the user is not sure of the severity of the dysfunction. If the user is aware, or at least has a subjective sense of how severe the dysfunction is, then this procedure can be skipped, as it can be time consuming, and cause an interrupt in the study. So in these cases, the severity scale reported by the user is noted in the form. In cases where it is obvious that the users' subjective rating is not corresponding with their condition (for example under or over-rating), then the test-personnel will try to identify the severity in accordance with the severity rating scale which was defined in section.

The application of the algorithms require a set of scale values to be used. The scale values determine the intensity of the algorithm, and as such must be selected carefully in order to not cause the test subjects to experience negative factors intentionally. The scales were obtained from several different papers, which were compiled into one large summary article published in a IEEE journal. In that paper, a 15 year recap was provided of redirected walking research, and results from estimating safe scale values were discussed.

For the rotation algorithm, a minimum and maximum value was selected to be within the range of the values in the article, but extended by a small margin. The steps between the minimum and maximum were not taken from the article, but were arbitrarily picked within the range. The rotation algorithm contains the following scales: 1.0, 1,1, 1,25, 1,4 and 1,5. The translation algorithm has the following scales: 1.0, 1,10, 1.15, 1.20 and 1,25 and the curvature algorithm has: -1.0, 5, 7.5, 10 and 12.5.[31]. The values were pushed slightly beyond the maximum in order to see if it was possible to stretch the scales or not.

3.7 Design of algorithm

The design of the algorithm consists of implementing the three redirection techniques and the reset mechanism. The reset mechanism is not a part of the algorithm itself, but is an extra safety measure that is used as users may move in unexpected ways in which a collision is unavoidable. The reset mechanic is supposed to interrupt the user and instruct them to return to a safe trajectory so that the actual test itself can be resumed.

Two important properties of the algorithm is the type, and the scale. The

---

[31] Niels Christian Nilsson et al. “15 Years of Research on Redirected Walking in Immersive Virtual Environments”. In: *IEEE Computer Graphics and Applications* 38.2 (2018), pp. 44–56. DOI: `10.1109/MCG.2018.111125628`.

type refers to how the redirection itself is accomplished, be it amplifying the head rotation by a scale, the displacement or subtly rotating the scene.

The second property is the scale. The scale is how strong the algorithm its placed. When it comes to algorithms which amplify the effect, then the scale typically refers to some value which multiplied by the base effect (be it head rotation or displacement) will result in the increased output. In the special case of the curvature algorithm, which is outlined below, then the scale refers to the radius of the circle in which the algorithm will try to keep you inside of, preferably on the circle perimeter.

3.7.1 Rotation algorithm

The rotation algorithm, as mentioned earlier is an algorithm that amplifies the rotation that the user performs in the virtual world. Suppose the user is wanting to move to an object in the virtual world. First they would have to turn themselves (rotate) towards the object, and then move towards the object. In this case, the rotation is multiplied by a specific scale. In this study, the scales used for rotation were 1.0, 1.1, 1.25 and 1.4. Suppose the scale 1.0 is picked. The user performs a rotation such that an object is in the center of their field of vision. If the rotation would require 45 degrees of rotation in the real space, then in the virtual world, the user would have to rotate 45 degrees as well. Since the rotation is multiplied by 1.0, which results in no effect.

Another example is if the user performs the same rotation, but the 1.4 scale is used. Again, if 45 degrees is needed, only a rotation of 32.1 degrees is required in the real world. This is because the rotation is amplified by the scale value. After the rotation is done, the user would have to move equally in the physical world as they would in the virtual world. Only the rotation is effected.[32]

---

[32] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. *Redirected Walking.* 2001.

The means to achieve this is by measuring the rotation which the headset is reporting as the user is rotating, and turning this into a Unreal Rotator object. The Rotator is broken into three components using the BreakRotator() function. The broken Rotator object has three properties, the yaw, roll and pitch. In this case the pitch and roll is not used. The yaw is multiplied by the scale value mentioned earlier, then re-assembled back into a Rotator object, then created into a transform object. The transform object has three inputs, the location, rotation and scale. In this case, the location and scale is reported normally, however the modified Rotator is given to the function, and the amplified rotation is applied.

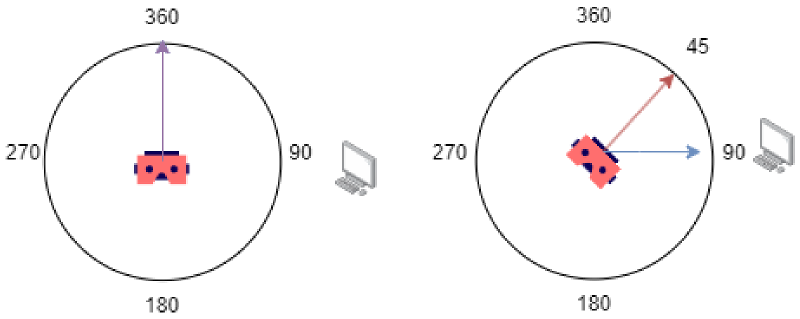


Figure 11: The user wishes to look at the computer on their right flank. The purple arrow is the overlap of red and blue arrows. Red arrow is the direction of the headset in the real world, and the blue is the direction of the view in the virtual world. Here the rotation scale is 2.0, doubling the rotation. The action only needs a 45 degree rotation in the real world.

3.7.2 Translation algorithm

The translation algorithm works in a similar fashion as the rotation algorithm. The algorithm works by observing the displacement in either the x or z axis. The displacement is then multiplied by the scale value, amplifying its

---

DOI: `10.2312/EGS.20011036`. URL: `https://diglib.eg.org/handle/10.2312/egs20011036`.

effect.[33] Since the test taker is not going to be traveling up stairs or heights, the y axis is simply reported normally. The reported data is then multiplied by the scale value.

Recall earlier how applying the rotation required supplying the Rotator object as one of the arguments for the Transform object constructor. One of the arguments for the transform object's constructor is a location parameter. The translation is applied in a similar fashion by multiplying either the x-axis or z-axis with the current scale which is to be applied, and then creating a transform object through the MakeTransform() constructor and afterwards applied.

Figure 12: The user wishes to approach the computer 4 meters ahead. The translation scale is 2.0. The user only moves 2 meters in the real world, and has been displaced 4 meters in the virtual world

3.7.3 Curvature gain

Curvature gain is applied in a more interesting manner, as the goal of the curvature gain is to subtly rotate the user so that they are traveling in a curved path which eventually forms a circle. One important difference between the scales is that for translation and rotation, the higher the value, the

---

[33] Mahdi Azmandian et al. "Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance." In: *ICAT-EGVE*. 2015, pp. 93–100.

more intense the effect. This is not the case for curvature, as the lower the value, the stronger the effect. Let the value n represent the scale. Suppose $n = 3$, then the curvature algorithm attempts to make the user follow a curve which will eventually form a circle with the radius of 3. If $n = 2$, then a smaller circle is made, which means that the curve in which the circle will be formed is much sharper. Therefore a lower, non-zero value represents a stronger effect.

The curve is expressed as a fraction of the radius. The expression $1/r$ where r represents the radius of the circle. Since the effect is applied per frame, it is necessary to know how much to apply. This is dependent on the velocity of the user v and how long a frame lasts T. First, one must calculate how many frames are needed in total for the entire curve which forms the circle. This can be obtained by calculating the circumference of the circle and dividing it by how long a frame lasts

$$S_{circ} = 2 * \pi * r$$
$$N_{frames} = \frac{S_{circ}}{T}$$

The amount of degrees rotation can be calculated by the following

$$degs = \frac{360VT}{2\pi r}$$

Every frame where there is movement, $degs$ rotation is applied to the user, so that eventually a circle of radius r is formed. The circle centre is always in the centre of the room, which is specified when the safety system is initiated in the headset. The degrees applied is either positive or negative, in the cases of the curve needing to be applied to the right, or to the left of the user in relation to the circle centre.[34]

---

[34] Emil Kjenstad et al. "Experiencing Rotation and Curvature Gain for Redirected Walk-

The entire goal is that the subtle left or right curving will let the user subconsciously correct themselves towards their goal, whether it is a button, a door or simply moving towards some point of interest, the user will correct themselves. At higher (lower) values, more space is required, but the detection thresh-hold will be higher according to previous research[35] and possibly the simulator sickness score lower, as physically, there is less of a rotating movement detected by the vestibular system.

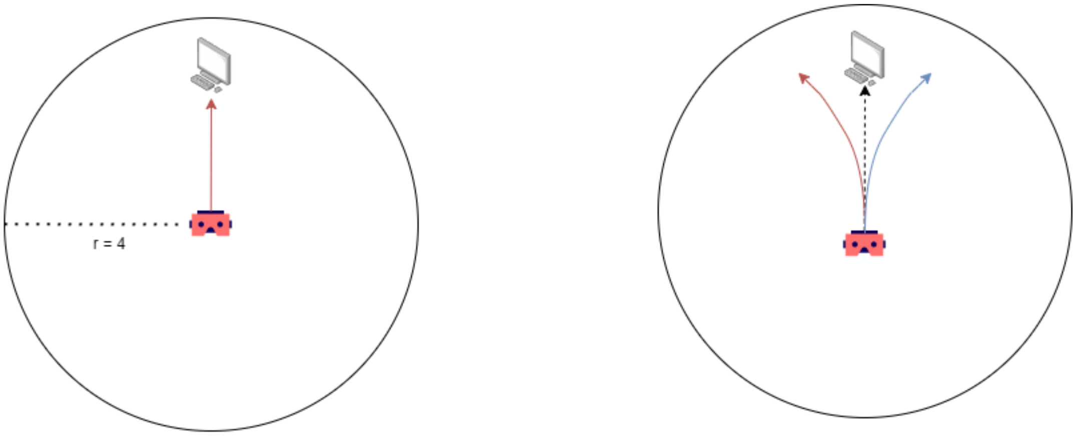


Figure 13: The striped arrow represents intended movement, while the red path shows the movement produced when applying the curvature. The blue path is the subconscious corrective effect. The user thinks they are walking straight, but are being tricked into correcting their path which leads into orbit around the circle

---

ing in Virtual Reality". In: *International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '23)*. Vancouver, BC, Canada: ACM, 2023. ISBN: 979-8-4007-0189-4/23/06. DOI: `10.1145/3592834.3592874`. URL: `https://doi.org/10.1145/3592834.3592874`.

[35] Timofey Grechkin et al. "Revisiting Detection Thresholds for Redirected Walking: Combining Translation and Curvature Gains". In: *Proceedings of the ACM Symposium on Applied Perception*. SAP '16. Anaheim, California: Association for Computing Machinery, 2016, pp. 113–120. ISBN: 9781450343831. DOI: `10.1145/2931002.2931018`. URL: `https://doi.org/10.1145/2931002.2931018`.

3.7.4 How is the effect applied?

In order for the effect to be applied in a timely manner, then it is not possible to wait and calculate the difference in displacement, and simply multiply the value and then apply it. The displacement needs to be multiplied and applied simultaneously as it is being performed. In order to facilitate this, a Unreal TickComponent is used to apply the effect. TickComponents are components which are called at the smallest unit of time which Unreal Engine 5 (or engines in general use), ticks. Ticks are based of the amount of frames rendered per second. If 60 frames are rendered per second, then there are 60 ticks per second. Such components are heavy on the Engine, as they are called every frame rendered, therefore it is important to not over-complicate the logic in the tick-component, as the component has a very heavy effect on the performance of the application.

Every tick, the TickComponent responsible for the redirection is executed. First the current actor rotation and location is gathered from the HMD. These are of types FRotation and FVector. In the case of rotation, the FRotation object is relevant, while the FVector is more relevant for the two other algorithms. This object is applied through the AddActorWorldOffset() function, which is an instance method of the Actor object. The Actor object is the object which represents the avatar of the user and its position in the world.

To summarize, through an instance method of the Actor-object, a function is called every tick which is every frame rendered. The reported object (from the HMD) representing the actors current position or rotation is broken down, the correct component is multiplied, and the irrelevant components are reported normally. The object is then packaged back into a transform, and applied through the AddActorWorldOffset() method of the actor-object.

3.8 Safety mechanism, "Resetting"

The algorithm cannot provide a safe and faultless redirected walking experience. The user can end up in situations, where they are about to exit the safe play area, which is predefined before the test is started.

In instances where the user is about to leave the safe area, a safety system is engaged in which the user cannot move. The user is warned that they are about to leave the safety area, and are given instructions to "reset" themselves so that they can return to a safe trajectory. The user is instructed to rotate themselves towards a highlighted point in the digital space. Once the user is aligned with the point, they are supposed to move towards that point. During the aligning process, the users view direction is paused in the virtual world, alongside their movement. By rotating with these two properties locked, the user is able to return to a safe position without changing their view or position in the virtual world.

It is based off of the "freeze-and-turn" resetting concept by Williams et al.[36] however certain modifications have been done to reduce the nauseating effects of the procedure. During the reset procedure, a vignette is applied to the users field of vision, making it hard to see objects not directly centered in the field of view. This is to make it difficult to spot objects in the distance, as focusing on these and performing the rotation movement in order to reset would become nauseating for the user. The vignette also draws attention to the instruction box which is shown in the middle, however this is merely a side-effect.

Ideally, the algorithm should work without the need for such a resetting procedure, however this requires the user to use a very large space so that

---

[36] Betsy Williams et al. "Exploring Large Virtual Environments with an HMD When Physical Space is Limited". In: *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization.* APGV '07. Tubingen, Germany: Association for Computing Machinery, 2007, pp. 41–48. ISBN: 9781595936707. DOI: `10.1145/1272582.1272590`. URL: `https://doi.org/10.1145/1272582.1272590`.

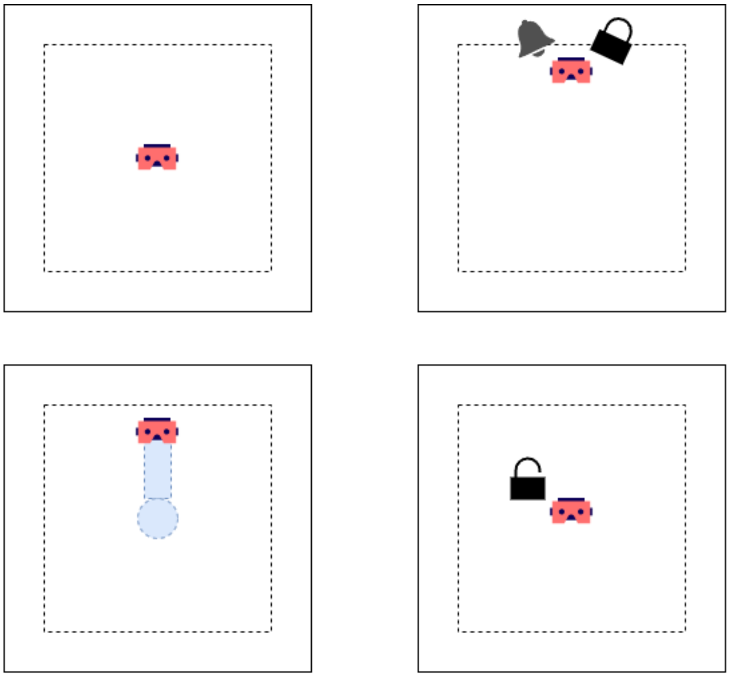


Figure 14: The reset mechanism in progress. First square: User is in safe state. Second square, user is too close. Position locks and is instructed to follow blue striped square to centre of room. Fourth square: User is in safe state and redirection resumes

resetting would never be needed. Nevertheless, it is a tool which combined with the algorithm can extend the play space and accessibility of redirected walking technology

3.9 Technical challenges during development

During the development of the system, several technical difficulties were met relating to the deployment of the experiment. The technical difficulties varied from engine-crashes due to the hardware present in the machine to, to engine-related technical gotchas, stuttering and loss off tracking during the pilot-test and main study. The instances were loss of tracking was experienced are not counted in the analysis section.

The first issue encountered, which was present since the start of the development cycle was a recurring graphics processor related crash in Unreal Engine 5. The error in particular was a DirectX 12 error occurring on graphics card. The study used a computer with a NVidia 4070 graphics card, which is a fairly new graphics card capable of handling heavy graphics workloads, such as rendering the scene in Unreal Engine 5, or running graphics heavy multimedia applications. Encountering such an error on a new and powerful graphics card was an unusual sight, and was not solved until the beginning of the testing stage. The error cause is still unknown, however it seems as if reducing the clock-speed on the graphics card remedied the issue. The downside is marginally worsened performance, which is outweighed by the benefits of not crashing. Other solutions were tried, such as changing the DirectX version to DirectX 11, however this messed with the rendering of the scene. This produced ”vaseline” smeared meshes, blurriness and artifacts. Changing the API completely to Vulkan was considered, but not tested as under clocking as a solution was discovered before changing the graphics API.

The second issue encountered was relating to the router used to connect the headset to the computer. This allows for a wireless connection through a proprietary protocol designed by Oculus (now Meta): AirLink. The exact specific of the protocol is outside the scope of this thesis, however it allows a headset to connect to a computer over standard 802.11 Wi-fi. The protocol has some limitations, such requiring a 5GHz network, however it will run on a 2.4GHz network, but at significant performance loss in the form of synchronized tracking. In order to connect the computer and headset through this protocol, a commercial 5GHz router was used. The router is a ASUS RT-AC86U running custom firmware.

The issue encountered was two-sided. First, interference from the various wireless access points were an major issue when it came to delivering a smooth and latency free wireless virtual reality experience. In order to rectify this issue, a Wi-fi spectrum scanner was used to select a channel that is not occupied by an access point or other device. The spectrum scanner was installed on an Android phone and obtained from the Android App-store. This allowed the test personnel to instruct the router to use a channel that was not occupied by any other wireless device based on the results from the spectrum analyzer.

The second issue was related to the range of the device. The router was placed 8 meters away from the computer and connected with a 30 meter long Ethernet cable. The remainder of the cable was coiled at the router and pulled around a pillar to create a straight path to one corner of the testing area, then to the router. This was to ensure that there were no trip hazards in the play area. The setup is illustrated in figure 15

The router was not able to keep up with the requirements of the Oculus Quest headset. In order to boost its performance, the transmission power of the router was increased from 25mW to 75mW. This provided a better connection, as at 25mW, the router could not keep up, and the tracking would be lost. This did however increase the wear on the router's compute and antennas. Lastly, the router channel width was set to the smallest possible value of 20MHz. Since only one device was to be connected, then there is no need for a wider channel for collision avoidance. It was set to the smallest value in order to optimize the router's cpu usage, especially considering the increase of transmission power.

With the above configuration, not a single test instance was interrupted due to the router not being able to keep up, and dropping frames. It is however important to note that frames may have been dropped, but not noticed by the user due to lack of experience in terms of how latency and performance is on a wired virtual reality headset, versus a wireless one.

The area that was used in the experiment was a open space of 8 by 11 meters. When considering replicating this setup in smaller areas, then having to modify the router's parameters to increase performance may not be needed.

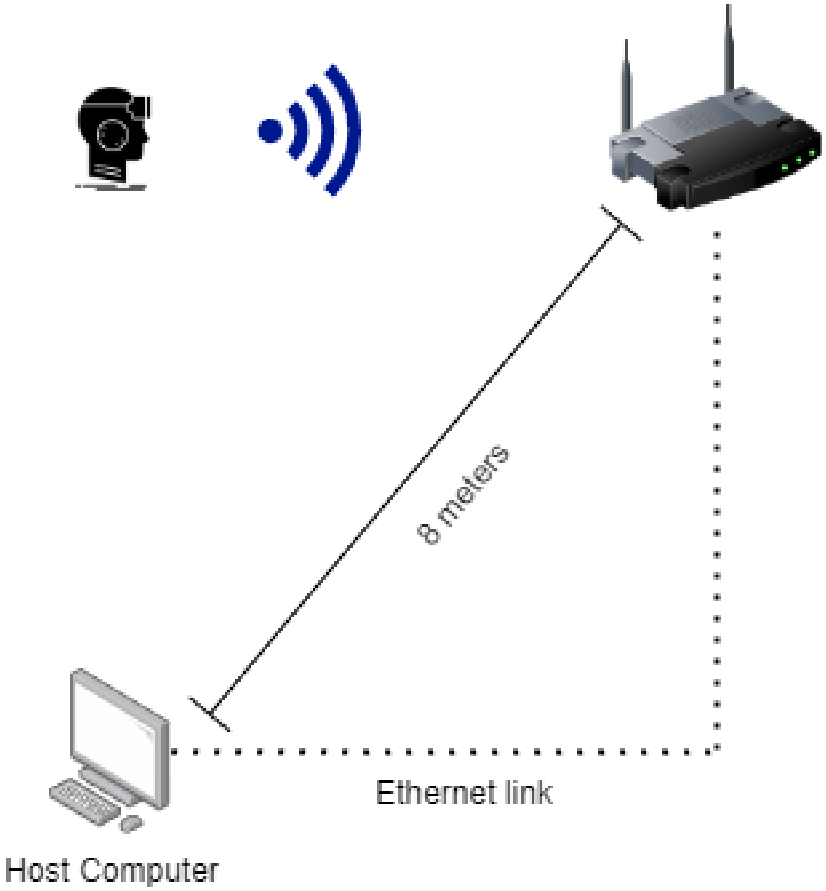


Figure 15: Illustration of the setup

For larger areas, it may be worth discussing an approach using several wireless routers in mesh or access point mode, as the strain on a single router to service the virtual reality headset could become too much. The router averaged at 62 degrees Celsius. Pushing the transmission power even more would cause damage to the router, and possibly be in violation of radio-transmission laws. Lastly, an alternative to using WiFi technology is to use a millimeter-wave technology as discussed earlier chapters, however this requires an adapter which increases headset weight.

The most important issue which was encountered was relating to how the Pawn and the Camera objects interact with each-other. In Unreal Engine, a pawn is an object representing a player or non-playable character. In general, anything which can be placed into the world is an actor, but the ones the player will control are pawns. The camera is a subclass of the Actor class. The camera is used to provide a view port into the world for a specific subset of actors, including pawns.

How Unreal Engine 5 handles movement in virtual reality is not as straight forward as one would expect. If the user physically moves in the real world, the pawn is not moved. Instead, the camera is offset by the displacement. When polling for the position of the actor, the same position is always returned, as the offset is only applied to the camera.

In order to rectify this issue, it was necessary to modify how OpenXR, which is the middle-layer between the virtual reality headset and Unreal Engine. The modification involved changing the behavior of the OpenXR function which reports the HMD movement, GetCurrentPose(). During the function call and before it terminates, a different function call is made to move the camera. After the camera is moved, an object containing x=0, y=0 and z=0 is created and passed as the return value for GetCurrentPose(). The value is then reported to Unreal Engine, and is interpreted as no movement being done. This essentially coupled the camera to the actor, and allows applying transformations without displacing the camera separately from the pawn.

4 Methodology

The entire study is divided into three sections. The pilot study, which was a study set before the main study in which the goal was to identify early pitfalls, problems and in general polish the system and survey before the main test. After the pilot study comes the main study, which included the the survey, with expanded questions for assessing simulator sickness, and a post-test game-play experience questionnaire which was used for developing acceptance testing (Outside the scope of this thesis).

4.1 Methodology Pilot Study

Before the full study, a pilot study was performed in order to identify technical errors which were not visible in the design phase of the study. The methodology in the pilot study is not the same as the full study, as several elements from the survey (described in detail in next section) were omitted. In addition the selection of VR-environments used were different from the main study.

The reasoning for this is to provide a minimal, clutter free environment to test out how well the users are able to move around in VR while they are being redirected. The suvery which was administered looks like the following:

- 1.1 Did you notice the redirection
- 1.2 How would you rate the experience from 1 to 5 where 5 is acceptable and 1 is not acceptable
- 1.3 How would you rate you sense of sickness, where 1 is not sick, and 5 is on the verge of vomiting.

The form is a minimal variant of the full study form, where input and critique from participants and on-lookers were used to further develop and expand on the survey. The data collected was used in analyzing the performance and acceptability criteria of the algorithm, hence why there only are three points which utilize a discrete numeric scale.

4.2 Methodology Full study

The goal of this thesis is to identify technical factors in an redirected walking setup which contribute to negative experiences; VR-sickness. In order to answer this question, data needs to be gathered in order to identify relevant factors, and dismiss irrelevant factors.

The data was collected from two sources. The first source is the HMD, and the second source is from a questionnaire which the users answer between each test, in addition to a post test 3 question section. In addition to these two sources, participants were asked to report basic relevant bio-metric values and subjective scores relating to how experienced they are with virtual reality, how experienced they are with video games, and how well the participant rates their own sense of balance.

4.2.1 HMD Data

The HMD gathers objective data about the users position in both the virtual and digital space. In addition to the aforementioned, it gathers data about what kind of redirection algorithm the user is being subjected to (gain) and what kind of scale (the intensity) is being applied. Below is a table of what data is being gathered.

- The HMD position in the real world
- The HMD rotation in the real world
- The HMD position in the virtual world

- The HMD rotation in the virtual world.
- What redirection algorithm is currently being applied
- What the current gain parameter is.

The objective data is primarily used to understand how well the redirected walking algorithm is working. It is divided into sensor data, data which relates to the hardware sensors and engine data, data which relates to the test level being ran in Unreal Engine. These terms are only for differentiating between two categories of objective data being logged.

For this thesis, the engine data is more important rather than the sensor data, as we do not ask the test taker to indicate whenever they feel sick. The downside with this approach is that it is not possible to investigate how the movements induced by the redirection could affect the sickness factors and their prevalence.

4.2.2 Questionnaire data

The second data source is an questionnaire which the user is asked to answer. The user does not take of the VR headset. Instead, the test administrators will read up the question and answer format and record the response.

The user is prompted after they finish each test. Not all questions are prompted after each test, this is to save time, as the average time to completion was approximately 30 minutes. Eventually the user will know what questions will be prompted, and can answer them without requiring the test administrator to read which question is being asked. Below is an example of the form without the SSQ matrix.

- 1.1 Did you notice the redirection
- 1.2 How would you rate the experience from 1 to 5 where 5 is acceptable and 1 is not acceptable

- 1.3 How would you rate you sense of sickness, where 1 is not sick, and 5 is on the verge of vomiting.
- The SSQ Matrix 3
- Was physical intervention required for maintaining safety? (Yes/No)

Symptom	Mild	Medium	Heavy
Dizziness			
Nausea			
Spinning head sensation(BPPV)			
Degraded Balance			
Worsened Sense of position (AVS)			
Throwing up/Feeling of throwing up			
Non critical loss of balance			
Critical loss of balance			

Table 3: The full SSQ matrix

When it comes to answering the SSQ, a matrix is provided to the test administrators. Which is identical to the one in 3. If the user answers a value greater than 1 for question 1.3 then the SSQ will also be asked. If the value is equal to 1, then the SSQ is not prompted, as the user does not feel sick.

The SSQ is normally applied for a single simulator instance. In this case a single simulator instance would mean one particular scale. This would produce very long testing times, since the simulation would have to end, switch to another gain, and then run again. To avoid such cumbersome swaps, the SSQ is asked after every test. The answers are then grouped together into three bins. The bins represent the three gains that are being tested. Redirection, Curvature and Translation gains. Inside the each respective bin, the scale is used as a key into the answers of the SSQ. The results of the SSQ is then used to compare the different scales in a single gain, and if applicable the gains against each-other.

In order to answer the question of which technical factors influence VR-sickness, statistical methods such as analysis of variance testing and Tukey's honestly significant difference are going to be used. The confidence intervals to be used for this study are 90% 95% and 99%

4.2.3 Participants

The participants were sampled from the students and faculty at the institute of informatics, as well as colleagues and friends of the test administrators. The participants were given a registration form which was filled before the test, and they were presented with a pre-test info brochure which described what types of tests they were taking, but not what the tests were actually testing for. The registration form is described below

- Age
- Height in centimeters
- Gender in Male, Female or Other option.
- Use of medication which may make someone more nauseous or less nauseous
- The use of optical corrective devices such as glasses or lenses
- Self-reported experience with Virtual Reality rated on a scale of 0 to 5
- Self-reported experience with Video games rated on a scale of 0 to 5
- Self-reported sense of balance rated on a scale of 0 to 5

4.3 Data processing

The data was obtained from two sources. The important point is to link the responses which is filed into the form, and to connect them to the objective sensor data which is reported from the engine. As there is no way to see what type of algorithm or scale is used just from the form. The output format from the headset is a .csv file, and the output from the form is a tab indented .csv file, which was converted to a Microsoft Excel .xlsx file, as the tab indentation is not compatible with pandas due to how it represents the sickness simulator questionnaire matrix.

The data is connected by a id-value which is obtained when the form is submitted. The id parameter does not contain any identifying information, and is only used to connect the form to the sensor data.

Afterwards, the form and sensor data is joined in a two-step join. First, the data is joined so that a dictionary is created, where the algorithms are the keys. The values are lists of dictionaries, where each sub-dictionary is indexed by the scale parameter. Inside the sub-dictionary, lies the actual answers for the Algorithm-Scale pair. This is the first way the form is linked. The actual object at the bottom of this structure contains the answers to the questions, and the type of algorithm and scale used.

The data is then processed in Python using numpy, pandas, scikit-posthoc library, matplotlib and seaborn for data visualization. Scikit-posthocs is used for the statistical tests, while numpy and pandas is used for reading in the data as a data frame, manipulating it and producing a better structured format for the posthocs test.

5 Results

5.1 Demographics

The participants in this study is composed of random volunteers sampled from the Institute for Informatics at the University of Oslo. The test setup is placed in a high-traffic section of the institute and users who wish to participate are asked to fill out a registration form that asks the following and are given a scheduled time to meet to perform the test.

The population of the Institute of Informatics (IFI) consists of computer science students, students associated with the psychology department who happen to be using the lecture halls and occasional passerby from surrounding institutes and businesses. There was no data-points related to where each participant was from, and as such the claim is only from anecdotes encountered during testing from talking with the participant.

The study contained 17 participants with the mean age of 26 years. The youngest participant was 19 years old and the oldest participant was 56 years old. The distribution of genders were 12 males and 5 females. No users reported the use of ”other” gender category. This equals a 41% female to 59% male ratio, which is representative of the institutes gender distribution which is reported to be 38% female to 61% male on bachelor-level.[37] Lastly, the mean reported sense of balance skill was 4 with two outliers at 5 and 3. Even though the demographic for the participants in the test resembles the demographic for the institute, it is important to note that not all participants were students or staff at the institute. Some members were friends or family asked to participate in the test. The similarity of the demographics are due to pure happenstance. The statistics can be seen in figure 16

SSQ Scores and symptom frequency

During the experiments, no critical symptoms were encountered. Symptoms that are critical are symptoms that would warrant termination of the test due to the safety of the test participant being greater. Negative symptoms and negative effects were encountered during testing and are reported in figure 17 and are not grouped by the gain or scale, they are a general report of which and how many were encountered. Two instances were recorded where the test was not able to be finished, due to a computer crash and one individual who had to terminate the test due to being late for a meeting. This number

---

[37] URL: `https://www.mn.uio.no/ifi/om/aktuelt/aktuelle-saker/2021/jentene-strommer-til-informatikkstudiene.html`.

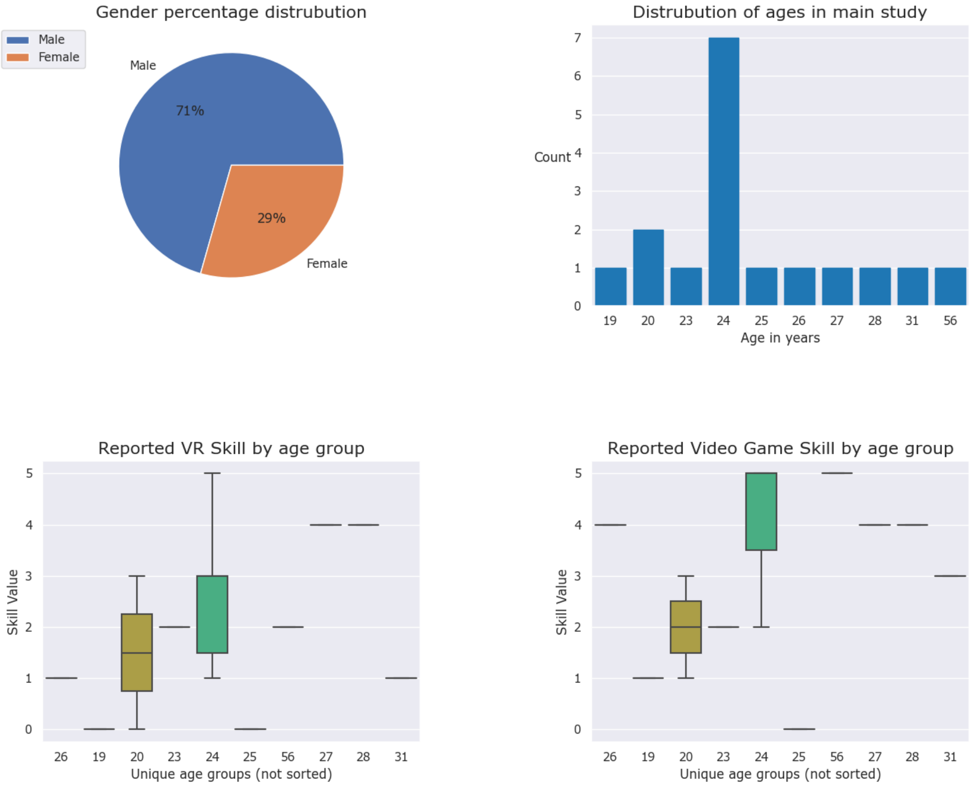


Figure 16: Basic participant statistics for full study

is reflected in the previous section.

The abbreviations "DGR_BAL", "AVS, "NCRIT_L\B", "CRIT_L\B" stand for "degraded balance", "Acute vestibular symptom", "non-critical loss of balance" and "critical loss of balance". They were shortened in figure 17 in order to save space for the labels. The figure does not take into account the varying severity degrees of each symptom category, only their rate of occurrence for the entire experiment.

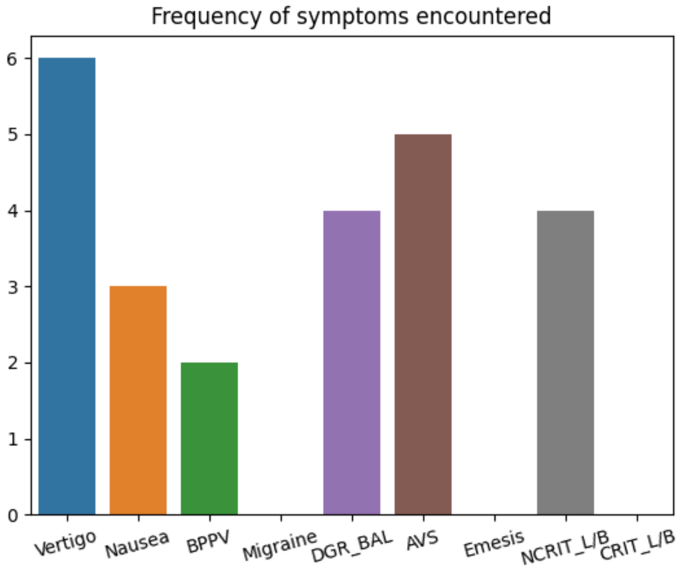
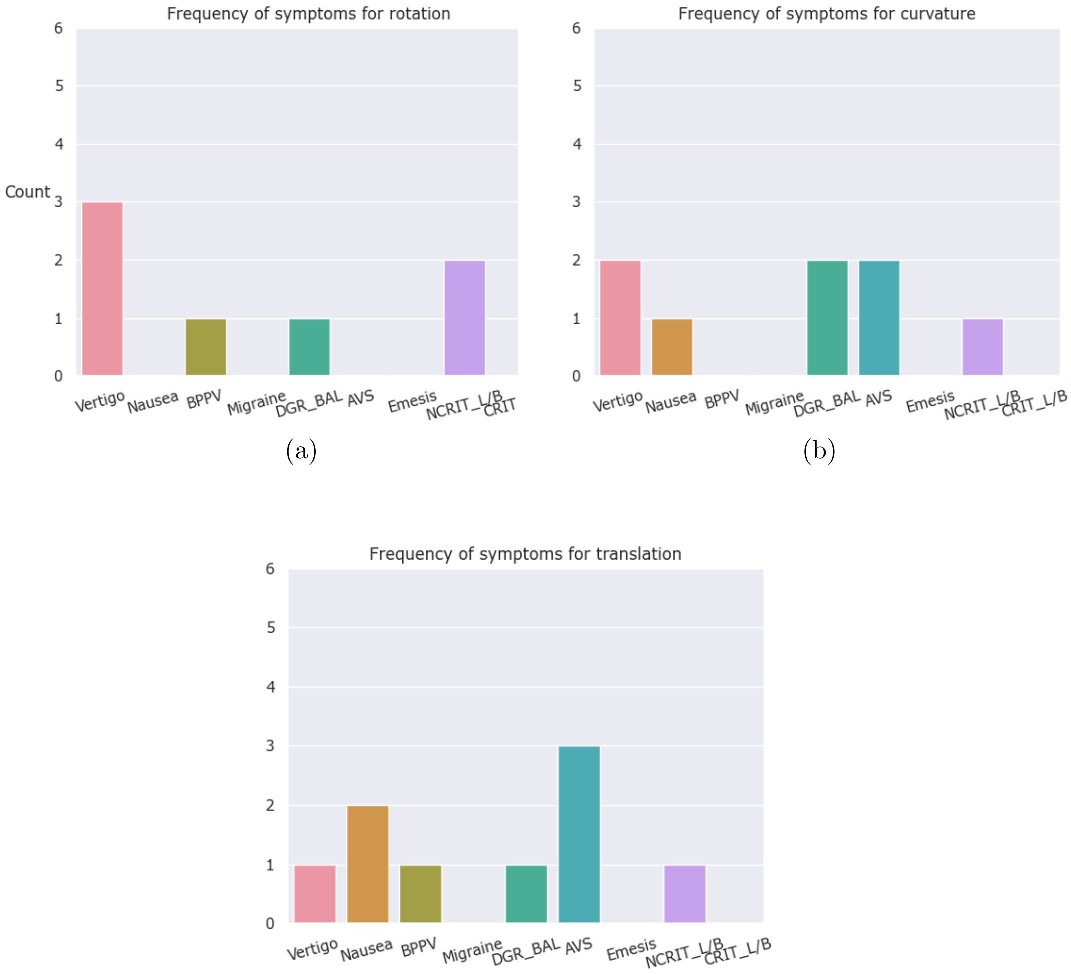


Figure 17: The amount of times a particular symptom was encountered

When placing the rate of occurrence for each symptom into respective categories, where the categories are the type of redirection algorithm used irrespective of whichever scale was used, the following three diagrams are produced.

The computed SSQ scores for each gain algorithm were computed in the same fashion as the SSQ example in the chapter "Motion sickness and redirected" walking. No weights were applied, and the SSQ scores is the sum of each



(a)

(b)

(c)

Figure 18: The frequency of symptoms per gain algorithm. a = rotation, b = curvature, c = translation.

the individual simulators (gains algorithms). The scores are illustrated in figure 19. The minimum score observed was 5 points for the rotation and the maximum points observed was 9 points for the translation algorithm. The higher the points, the worse of a simulation it was in relation to the scoring system which can be observed in figure 8.

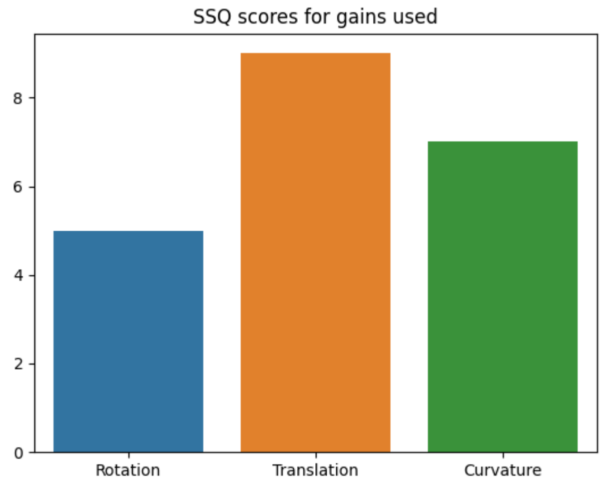
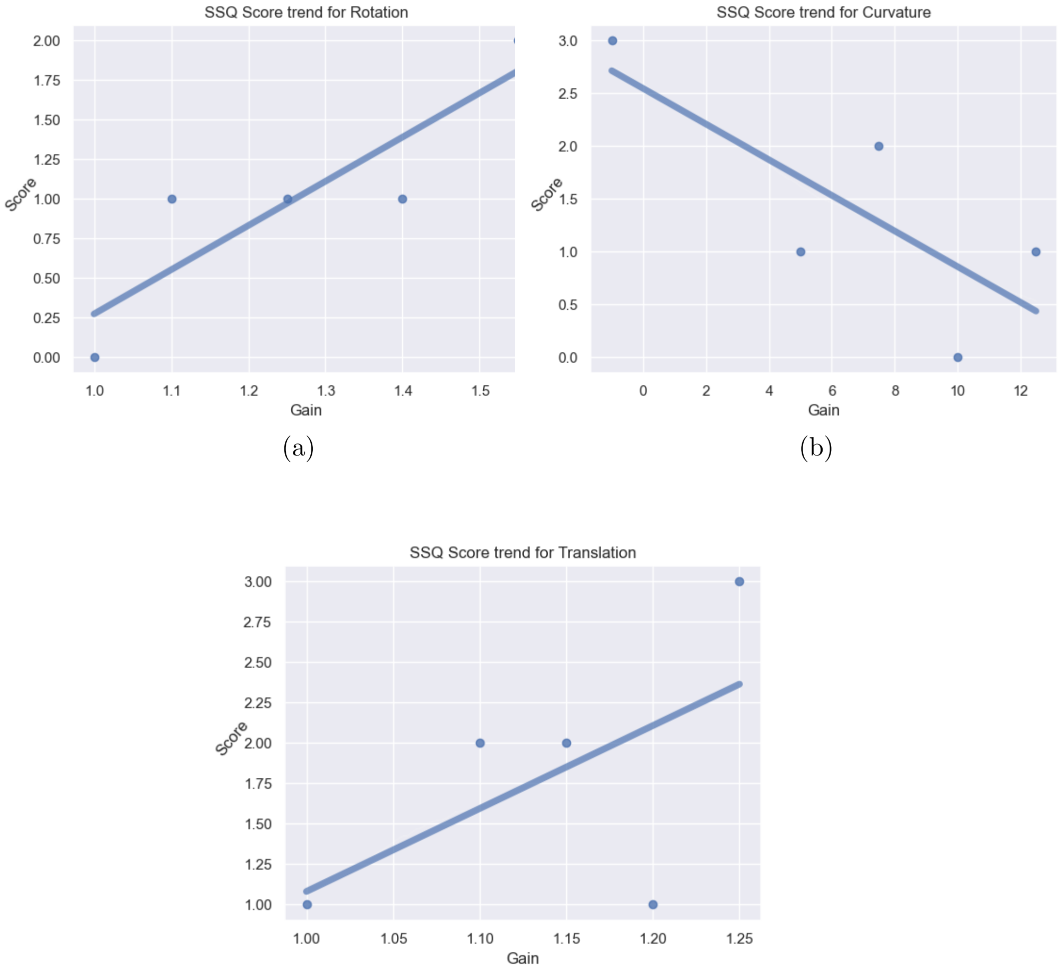


Figure 19: The SSQ scores for each gain

In addition to the computed SSQ scores for each algorithm, the SSQ score values per algorithm is illustrated. This can be observed in figure 20a, 20b and 20c. The y-axis represents the ssq score, and the x-axis represents the scale selected for that particular algorithm. Only discrete points were tested, and not values in-between each point



(a) (b)

(c)

Figure 20: The SSQ score trend per scale for each gain that was used in the study. The dots illustrate the data-points for the gain. Trend-line represents linear fit. NOTE: The lower the gain value on curvature, the more intense the effect.

5.2 Statistical Tests on technical factors

Certain abbreviations have been used for the different gain algorithms that have been used. RT stands for "Rotation", CR stands for "Curvature" and TR stands for "Translation"

Applying the ANOVA test to the data set, which is separated into three bins representing the type of gain used; rotation, translation and curvature. The reported p values extracted from the ANOVA test is as follows

Axis	Rotation	Curvature	Translation
Rotation	n/a	0.506	0.26
Curvature	0.506	n/a	0.715
Translation	0.26	0.715	n/a

Table 4: Results of ANOVA test when testing against other algorithms. n//a indicates comparison against self and has no value. CI = $90^{*}, 95^{**}, 99^{***}$

The application of the ANOVA procedure on table 4 compares the SSQ score produced by one algorithm, against the other algorithm. The fields "n//a" indicate that it is a comparison from one algorithm against itself, therefore not a valid comparison. The other comparisons, where there is a valid value in a cell indicates a comparison from the header on the x-axis and the header on the y-axis. The goal is to see if any algorithm distinguished itself by outputting statistically significant value. If this was the case, a deeper investigation was to be performed.

The Tukey's honestly significant difference is also applied to see where the differences lie in terms of the comparisons, and also to produce the full set of values that are reported from the test. The full set of values include inverse tests.

The leftmost column that indicates the comparison is a comparison between the algorithms. For example (RT - TR) means that the Tukey's test was performed with the SSQ scores from the rotation gain algorithm versus the

Comparison	Statistic	p-value	Lower CI	Upper CI
(RT - TR)	-0.065	0.260	-0.177	0.048
(RT - TR)	0.065	0.260	-0.048	0.177
(CR - RT)	0.040	0.506	-0.079	0.159
(CR - RT)	-0.040	0.506	-0.159	0.079
(CR - TR)	-0.024	0.715	-0.157	0.108
(TR - CR)	0.024	0.715	-0.108	0.157

Table 5: The full set of values performing Tukey's test. TR = Translation, CR = Curvature, RT = Rotation

translation gain algorithm.

If the largest and lowest values within each scale was to be tested, regardless of what gain algorithm they belonged to, then the following table would be produced

Comparison	Statistic	p-value	Lower CI	Upper CI
(RT 1.0 - RT 1.55)	-0.154	0.170	-0.378	0.071
(CR -1.0 - CR 10.0)	0.250	0.160	-0.106	0.606
(TR 1.0 - TR 1.25)	-0.147	0.336	-0.458	0.163

Table 6: The reported p values when comparing scales within gains against the value which indicates that the algorithm was not applied (1.0 for RT and TR and -1.0 for CR)

If a full factorial comparison is performed within each gain, meaning that a single scale is compared against all scales within that gain the following table is produced. Note p-values higher than 0.3 have been filtered to make the table more compact. Without filtering, the table would be 40 comparisons long

Comparison	Statistic	p-value	Lower CI	Upper CI
(R10 - R1.15)	-0.154	0.170	-0.378	0.071
(CR -1.0 - CR10.0)	0.250	0.160	-0.106	0.606
(CR -1.0 - CR7.5)	0.167	0.136	-0.056	0.390
(CR10.0 - CR12.5)	0.125	0.210	-0.077	0.327
(TR1.2 - TR1.25)	-0.154	0.296	-0.451	0.143

Table 7: Filtered results from full factorial Tukey testing. p-values greater than 0.3 have been filtered

6 Discussion and findings

In this section, the data that was reported in the previous section is going to be weighed up against the research questions which were stated in the beginning of this thesis. The questions are as follows.

- **RQ1:** How does the scale parameter in an redirected walking algorithm influence the prevalence of VR-sickness symptoms in an redirected walking setup?
- **RQ2:** How does the choice of redirection technique affect the prevalence of VR-sickness symptoms in an redirected walking setup?
- **RQ3:** What factor or combination of factors cause VR-sickness in redirected walking setups?

6.1 The choice of redirection technique and its effects on VR-sickness

The discussion in this section relates to comparing the redirection techniques, and their effects on VR-sickness symptom prevalence through the simulator sickness questionnaire scores. The three algorithms as mentioned throughout the paper is the rotation, translation and curvature algorithms.

The analysis of variance test performed in section 5.2 showed no statistically significant difference when it comes performing an analysis of variance test on the three groups. The groups were tested against each other, and inverse comparisons were not performed, as they produced the same p-value compared to the non-inverse comparison. The lowest p-value that was reported was 0.26 and the highest reported p value was 0.715. The values are not significant enough when comparing it to the selected confidence interval of 90*, 95* and 99* denoted by single, double and triple asterisk marks respectively.

While no statistically significant observation was present, which means that the null-hypothesis cannot be rejected, does not mean that the entire investigation into whether the redirection algorithm itself is an important factor is to be called off.

Comparing the algorithms on the basis of their simulator sickness scores, as shown in figure 19 shows that the simulator sickness score is different for each algorithm. The rotation algorithm scored 5 points, curvature scored 7 points and translation scored 9 points. The interesting observation is that comparing the scores between rotation and translation results in a difference of 4 points, one point short of twice the simulator sickness score. The distance in score for curvature to rotation and translation is -2 \\+2 points respectively. It is an interesting, however minor observation in which no real conclusion can be drawn.

The scores do not follow the exact simulator sickness score questionnaire scoring algorithm, in which the symptoms that were present are grouped into three categories: nausea, oculomotor and disorienting factors. Each section has a different weighting which is applied to produce the final score. The reason for this is to avoid inflating the scores as mentioned in an earlier section[38] and due to the slightly modified SSQ would require recomputing the weightings for the modified SSQ.

---

[38] Stéphane Bouchard, Geneviève Robillard, and Patrice Renaud. *Simulator Sickness Questionnaire–French Revision.* 2007. DOI: `10.1037/t73606-000`. URL: `https://doi.org/10.1037/t73606-000`.

When comparing these three scores to the a followup paper to Kennedy's simulator simulator sickness questionnaire paper, a categorical scoring metric is devised which describes the degree of suitability of the simulator[39]. The categories are separated with score ranges, where higher score is worse. The scoring is as follows:

Score	Categorization
0	No symptoms
<5	Negligible Symptoms
5-10	Minimal Symptoms
10-15	Significant Symptoms
15-20	Symptoms are a concern
>20	A bad simulator

Table 8: The categorization according to Kennedy

Trying to place the scores produced in 19 into the categorization leads to all of them being placed in "Minimal Symptoms" section. Rotation and curvature algorithms are placed firmly in the start and middle of the range, while translation is almost at the intersection of the minimal symptom category and significant symptom category.

To summarize, when comparing the algorithms against each-other, no statistically significant result is produced. The lowest score produced was 0.26 and highest score produced was 0.715. This can be seen in 4. It may be more fruit full to take a look at the scale parameter instead of the algorithm

---

[39] Kay M. Stanney, Robert S. Kennedy, and Julie M. Drexler. "Cybersickness is Not Simulator Sickness". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 41.2 (Oct. 1997), pp. 1138–1142. DOI: `10.1177/107118139704100292`. URL: `https://doi.org/10.1177/107118139704100292`.

6.2 How does the scale parameter in an redirected walking algorithm influence the prevalence of VR-sickness symptoms in an redirected walking setup?

The scale parameter as discussed earlier in this thesis is the value which represents the effect of the redirection.

The scores for the curvature algorithm shows an interesting observation. Recall earlier how the value -1.0 represents no effect from the algorithm. Refer to figure 20b. The score peaks at a simulator sickness score of 3 at -1.0, where no effect is to be applied. The reasons why this phenomenon is observed is unknown.

One argument is the learning effect taking place. Suppose one gets used to a scale, then arrives at a scale which no effect is applied. This might cause some nauseating effects which makes the user report that they felt sick. The obvious counter argument to this case is the randomization of the scales. The tests were given in a random order, and therefore it should not be possible to attribute the spike to the test takers to having learnt the effects. Therefore it seems more likely that the spike is simply due to error in the reporting from the users.

The curvature algorithm shows some interesting behavior considering it the properties it exhibits. Since the scale values relate to the radius of a circle, then it is possible to determine how much space one needs beforehand. It allows a end-user to clear out furniture out of a space, measure the area required, and use the headset with the specific curvature algorithm. By looking at the scores in 20b, despite going from a circle with radius 12.5 meters to a circle with the radius of 5.0 meters, which is a 2.5 times reduction in radius size, the simulator sickness score does not increase. It either increases by 1 point or decreases by 1 point. Considering this, it may be possible to push the curvature algorithm even further by halving the smallest circle that was

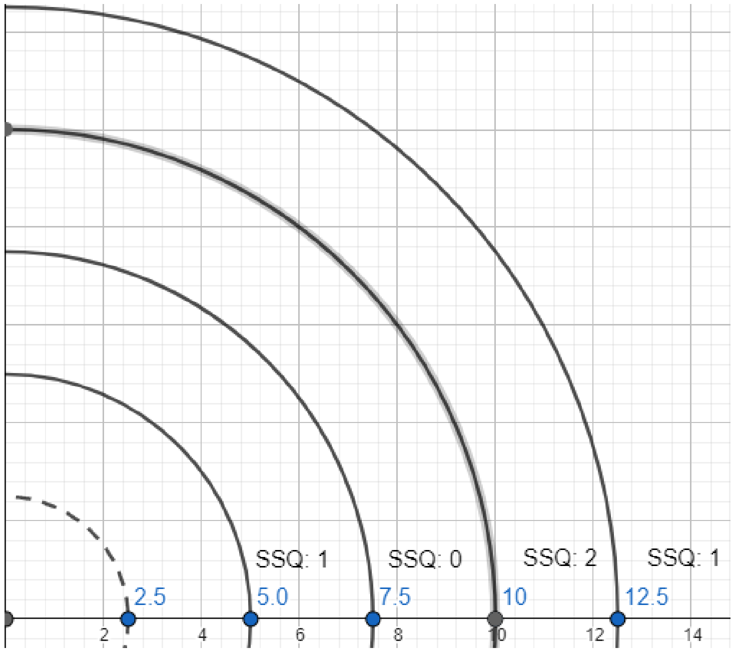


Figure 21: The distance from the origin to the intersection of the curves (for each respective scale) represent the radius of the circle which the user has to travel. The dotted half circle represent a possible goal for further testing. The goal of this figure is to illustrate the circular path over the sickness score

tested, and possibly produce still reasonably safe results. Safe results in this case would be having a score lower or equal to the ”Minimal Symptoms” range.

If a circle of 2.5 meters radius were to be produced, then this would represent a 5 times smaller circle than the largest circle tested, 12.5. With the score only varying by +/- 1 point, and with a conservative estimate of the score increasing by 5 times, then the score would fall in the 5-10 range. It would still fall in the minimal range with the conservative estimate. There is no evidence at all to support this estimate, but it creates a interesting scenario to explore, as a circle of 4 meters diameter is much more achievable for users at home, than a circle with diameters of 10 meters. Figure 21 shows how these circles would look like when compared to each other. The labels indicate the scale used, and the simulator sickness score produced by the scales. The dotted circle represents a useful goal when it comes to further pushing the curvature algorithm, will still possibly keeping the simulator sickness score in a safe range. However approaching the idea of reducing the circle radius must also be met with higher safety considerations, as there has not been any research on scales set this low.

For the rotation and translation algorithm's scales, in relation to **RQ1** it seems as if the greater the scale, the greater the chance of encountering vr-sickness symptoms. The claim is supported by observing the trend-line of both the algorithms, illustrated in figure 20a and figure 20c, which is intuitive considering the higher the scale, the more intense the redirection. For the rotation algorithm, this means that the rotation is greater. The user is turned a higher amount of degrees per frame. The translation algorithm is not expressed in terms of degrees per some time unit, instead the translation algorithm will just amplify the displacement in the real world, translated into the virtual world. For both these algorithms, the greater the scale, the higher the SSQ score. While no statistically significant observation was done, which can be seen in figure 6 and in 7 it seems as if comparing the largest values, to the smallest values, which are the ones where no effect seems to be applied results in lower p-values than comparing values close to each other. The greater the difference in scale, the smaller p-value is reported. While this is no definitive proof, it sets the stage for doing more research into how the scale itself effects rather than the algorithm itself.

A useful observation is the very high ssq score produced by the translation algorithm. The translation algorithm, on the basis of the simulator sickness score could be used to support the case that this algorithm is not good to use for people who wish to avoid the effects. The reason why translation produces such a higher score is not known. Some theories could be that there is some disconnect felt by the user when they physically move their legs or a disconnect with the scale of objects in the virtual space, versus the intuition of scales in the real world.

Suppose one wishes to take one step forward. The distance of the step might be intuitively known by the body, as it is a important bio-mechanical property, also known as length of the stride. When the eyes observe the displacement; which is the increased displacement in the virtual world, and the somatic system; the part of the nervous system which an individual can control, reports that the stride completed, then if there is a mismatch, then issues can occur. As the feedback from the somatic system; more precisely the nerves in the legs which report contact with the ground and the eyes work together to ensure effective locomotion. This cooperative effect can easily be observed when walking on stairs where the step size is suddenly changed, while sickness will not occur, a sharp disconnect is found because the body anticipated the stride to impact, but never happened in the expected time.

6.3 What factor or combination of factors cause VR-sickness in redirected walking setups?

The third research question, **RQ3** which states which factor or combination of factors cause VR-sickness in redirected walking setups is about whenever it is the algorithm, scale or both which are leading causes for VR-sickness or not.

When taking a look at both **RQ1** and **RQ2**, then it seems that both the choice of redirection algorithm, such as curvature and the intensity has some weighting into cause of VR-sickness in redirected walking setups. However, which of the factors, either the intensity or the algorithm choice is more important for VR-sickness?

It is difficult to determine which factor contributes more in relation to VR-sickness. There is not enough evidence to determine which factor is more important for the causes of vr-sickness in a redirected walking approach, as the statistical tests showed no statistically significant difference.

The conclusion to the question regarding **RQ3** is that there is not enough evidence to support that one of the factors weigh more when considering the occurrence of VR-sickness. Some equal weighting of the choice of algorithm and factors could be assumed, but ultimately the conclusion cannot be drawn.

6.4 Regarding the hypothesis

The hypothesis that was formulated is as follows:

- Hypothesis: The scale parameter is the only factor that produces a statistically significant increase in VR-sickness symptoms. The choice of redirection algorithms: curvature, translation and rotation do not impact the frequency of vr-sickness symptoms

The hypothesis states in practice that (if it were true) that we could disregard the algorithm and only focus on the scale parameter as the deciding factor that produces a *statistically significant* result.

From the data presented in chapter 5 (Results), then it is not possible to support the hypothesis on the basis that there isn't enough data in favor of the hypothesis to validate it. While the Tukey-test used to compare the scales against the control scales (the values where no redirection is applied), produces lower p-values close to the threshold set for this thesis, merely the proximity is not enough evidence to show the hypothesis to be true.

6.5 On the topic of mitigating effects

It is easy to point out a problem, however it is more difficult to propose a solution, let alone a good solution. The previous section highlights which factors could affect VR-sickness, however there is no clear answer on which factor should be focused more when it comes to reducing VR-sickness. The last paragraph pointed out that it could be more fruitful to focus on the scale parameter, and not the algorithm itself in order to minimize VR-sickness

Suppose one finds out what combination of factors affect VR-sickness. Are there any ways to minimize the VR-sickness? During the study, several interesting behaviors were observed by the test-takers which, when casually asked by the test-administrators why the test-taker participated in the unusual behavior, the answer was typically along the lines of "Helps me keep myself steady/makes the dizziness go away". Measures that minimize instability and vertigo, are important due to the fact that the most common reported symptoms were vertigo followed by AVS, which contains vertigo as one of the diagnostic criteria. In addition, most of the symptoms affect the vestibular system, which when disturbed is likely to produce nauseating effects.

The behavior that was observed was a test-taker holding their hands in front of their face with a little distance. Imagine a pugilists' stance. The hands are held in front of the face in order to provide a defence against the opponent. One participant showed similar behavior, due to their background as a martial arts practitioner. When questioned about the unusual behavior, the participant responded with "Having my hands in-front of me like this gives me a reference point, which makes me feel less sick".

The user has a limited avatar in the digital world, where the hands in the virtual world are represented by the a digital model of the controllers. This is done in order for the user to know where their hands are, in order to pick things up and interact with the environment. The concept of a reference point being used in VR is not new. It has been done in sit down VR experiences, such as VR enjoyment rides. The difference is however they usually are implemented as VR "noses", a model that is connected to character[40].

---

[40] Carolin Wienrich et al. "A Virtual Nose as a Rest-Frame - The Impact on Simulator Sickness and Game Experience". In: IEEE, Sept. 2018, pp. 1–8. ISBN: 978-1-5386-7123-8. DOI: 10.1109/VS-Games.2018.8493408.

The efficacy of such an addition into redirected walking is an interesting approach, as mitigating VR-sickness symptoms using this technique represents a easy to implement addition. In Unreal Engine 5. Such an addition could be made in less than an hour using the modeling tools to design the nose and attaching it to the avatar using mesh merging. One could also forego the bespoke approach. The approach of using a VR-nose for redirected walking could be substituted for detailed hand models instead, as digit-tracking is a feature that newer virtual reality controllers are employing, and Unreal Engine 5 ships with a built in example of digit tracking into a set of VR hands for use in applications.

Another observation which was done is that users reported that focusing on far away phenomena, may it be objectives, text screens or non playable characters in the world made them more aware of the rotation being applied, and during the reset procedure, they reported worsened experience in the form of dizziness. The reason behind why they suffered increased effects were due to the rotation being applied at twice the rate, so that the reset procedure was to work correctly. The user would rotate 180 degrees in the real world, but 360 degrees in the virtual world. This means that the user would be oriented in the opposite direction in the real world, but in the virtual world the user is facing the same direction as they did before the rotation. The users would not notice the increased rotation when focusing on objects close, but they would when they focused straight ahead in the un-darkened section of their field of view.

By combining the virtual nose approach, or in general, a close reference point to the user by having the virtual hands, it may be possible to mitigate the sickening effects that arise when focusing on a point far away, as the user would be asked to focus on the nose or keep their hands in front of them. It could even be combined with an in-game mechanic as to reduce the immersion break by having the user perhaps cast a spell by holding their hands up, and spinning slowly 360 degrees in the virtual world (180 in real life). In order to act as a distracting element, a light or flare could be placed at their finger tips as they perform this procedure. Sadly, this approach might fit only a select few modalities of virtual reality entertainment, such as fantasy settings.

The curvature algorithm, as mentioned, shows potential in being a redirection algorithm where the space required can be pre-planned, due to its steer-to-orbit nature letting a user know beforehand how much space is needed, but also due to how the simulator sickness score is scaling. Previous research seems to suggest that the sickness score can be brought down even further by asking the user to move a bit more slowly.[41] There needs to be more research on the algorithms by themselves in order to better understand their effects on the human body and how they may be deployed safely.

There are many ideas and techniques which have shown promise, and some whom have not been tested. As such there are many approaches which can be taken in order to mitigate the negative effects which users experience in not only virtual reality, but a redirected walking setup. Future testing is needed in order to both identify the factors, and then to mitigate them.

7 Future Work

Many questions remain unanswered relating to the fields described in **RQ1**. Especially considering how far one can push the scale parameter in a redirected walking system. Higher scales means better performance for the algorithms in smaller and more confined spaces, however for rotation and translation, it seems as if the higher the scale, the worse the simulator sickness score, which by proxy, could be said that the experience becomes worse. Where the limit in which an experience is still playable, despite the negative effects is not known, and further testing on higher scales is necessary in order to better

---

[41] Christian T. Neth et al. “Velocity-Dependent Dynamic Curvature Gain for Redirected Walking”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.7 (2012), pp. 1041–1052. DOI: `10.1109/TVCG.2011.275`.

determine who is suited for redirected walking experiences in terms of how much space they have available, how suited they are on a individual basis (for the user); for example if older people tend to have better experiences on certain scales, if the proficiency impacts the SSQ score or not and other factors.

In the previous chapter, the concept of having a reference point as a mechanism for reducing the effects of VR-sickness was mentioned. This is something which needs further investigation for the redirected walking concept, as it has shown fruit-full results in the sit-down virtual reality experiences, and it may be possible to replicate the effect in redirected walking. If so, this could allow higher scales, which offers better performance in smaller physical spaces, and in turn makes redirected walking much more accessible for individuals without large open spaces to their disposal.

Conclusion

The goal of this thesis is to identify which technical factors in a redirected walking implementation cause unpleasant effects such as sickness and balance, amongst others. The result of this thesis did not identify single factor with a very high degree of confidence, however the results seem to indicate that the scale of the algorithm plays a bigger role than just the algorithm itself, by virtue of statistical analysis in the form of analysis of variance testing (ANOVA).

In addition, several behaviors were observed that can be implemented in future endeavours to perhaps aid in reducing the effects. Features such as a digital reference point in the form of a nose or having your virtual hands reflect your actual hands through digit tracking. Such a feature is supported in many virtual reality headset packages.

While the focus on the study has been exploring virtual reality in terms of entertainment, it is also possible to take the findings and implement them in other applications, such as training, quality testing of subways, (by placing the user in a virtual ride with real height data) or for performing HMS simulations without renting out an expensive building in order to simulate the exact same dimensions and space as the real building.

Redirected walking is a exciting field, however the problems surrounding cyber sickness is an important challenge to tackle. Currently, as it stands, there are mitigating factors which should explored in order to create a more stable and safe experience for virtual reality interested individuals. If redirected walking wants to blossom into an immersive technology which can be safely deployed in a whole array of use cases, such as HMS-training, exposure therapy in psychiatry[42], for defence applications, or simply for fun and entertainment purposes. Then more investigation must be done in in the realm of how far the scales can be pushed to see if it is possible to tip the scales in favor of redirected walking.

References

[1] Arbeids- og inkluderingsdepartementet. *Lov om arbeidsmiljø, arbeidstid og stillingsvern mv. (arbeidsmiljøloven).* `https://lovdata.no/lov/2005-06-17-62/4-1`. 2022.

[2] Joseph J. LaViola. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32 (1 Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: `10.1145/333329.333344`.

[3] Anh Nguyen and Andreas Kunz. “Discrete scene rotation during blinks and its effect on redirected walking algorithms”. In: ACM, Nov. 2018, pp. 1–10. ISBN: 9781450360869. DOI: `10.1145/3281505.3281515`.

---

[42] Debra Boeldt et al. “Using Virtual Reality Exposure Therapy to Enhance Treatment of Anxiety Disorders: Identifying Areas of Clinical Adoption and Potential Obstacles”. In: *Frontiers in Psychiatry* 10 (Oct. 2019). DOI: `10.3389/fpsyt.2019.00773`. URL: `https://doi.org/10.3389/fpsyt.2019.00773`.

[4] Hiroaki Sakono et al. "Redirected Walking using Continuous Curvature Manipulation". In: *IEEE Transactions on Visualization and Computer Graphics* 27.11 (2021), pp. 4278–4288. DOI: `10.1109/TVCG.2021.3106501`.

[5] Eike Langbehn, Paul Lubos, and Frank Steinicke. "Evaluation of Locomotion Techniques for Room-Scale VR". In: ACM, Apr. 2018, pp. 1–9. ISBN: 9781450353816. DOI: `10.1145/3234253.3234291`.

[6] Michael Rietzler et al. "Rethinking Redirected Walking: On the Use of Curvature Gains Beyond Perceptual Limitations and Revisiting Bending Gains". In: IEEE, Oct. 2018, pp. 115–122. ISBN: 978-1-5386-7459-8. DOI: `10.1109/ISMAR.2018.00041`.

[7] Tobias Feigl et al. "Acoustical manipulation for redirected walking". In: ACM, Nov. 2017, pp. 1–2. ISBN: 9781450355483. DOI: `10.1145/3139131.3141205`.

[8] Young Youn Kim et al. "Characteristic changes in the physiological components of cybersickness". In: *Psychophysiology* 0 (0 Aug. 2005), 050826083901001–??? ISSN: 0048-5772. DOI: `10.1111/j.1469-8986.2005.00349.x`.

[9] Clare Regan. "An investigation into nausea and other side-effects of head-coupled immersive virtual reality". In: *Virtual Reality* 1 (1 June 1995), pp. 17–31. ISSN: 1359-4338. DOI: `10.1007/BF02009710`.

[10] Jann Philipp Freiwald et al. "The cybersickness susceptibility questionnaire". In: ACM, Sept. 2020, pp. 115–118. ISBN: 9781450375405. DOI: `10.1145/3404983.3410022`.

[11] Carolin Wienrich et al. "A Virtual Nose as a Rest-Frame - The Impact on Simulator Sickness and Game Experience". In: IEEE, Sept. 2018, pp. 1–8. ISBN: 978-1-5386-7123-8. DOI: `10.1109/VS-Games.2018.8493408`.

[12] S. J. Alexander et al. "Studies of motion sickness: XVI. The effects upon sickness rates of waves of various frequencies but identical accelera-

tion." In: *Journal of Experimental Psychology* 37.5 (1947), pp. 440–448. DOI: `10.1037/h0063240`. URL: `https://doi.org/10.1037/h0063240`.

[13] Robert S. Kennedy et al. "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness". In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: `10.1207/s15327108ijap0303\_3`. eprint: `https://doi.org/10.1207/s15327108ijap0303_3`. URL: `https://doi.org/10.1207/s15327108ijap0303_3`.

[14] Hyun K. Kim et al. "Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment". In: *Applied Ergonomics* 69 (2018), pp. 66–73. ISSN: 0003-6870. DOI: `https://doi.org/10.1016/j.apergo.2017.12.016`. URL: `https://www.sciencedirect.com/science/article/pii/S000368701730282X`.

[15] Pauline Bimberg, Tim Weissker, and Alexander Kulik. "On the Usage of the Simulator Sickness Questionnaire for Virtual Reality Research". In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2020, pp. 464–467. DOI: `10.1109/VRW50115.2020.00098`.

[16] Stéphane Bouchard, Maxine Berthiaume, et al. "Arguing in Favor of Revising the Simulator Sickness Questionnaire Factor Structure When Assessing Side Effects Induced by Immersions in Virtual Reality". In: *Frontiers in Psychiatry* 12 (Nov. 2021). DOI: `10.3389/fpsyt.2021.739742`. URL: `https://doi.org/10.3389/fpsyt.2021.739742`.

[17] Peter Johns and James Quinn. "Clinical diagnosis of benign paroxysmal positional vertigo and vestibular neuritis". In: *Canadian Medical Association Journal* 192.8 (Feb. 2020), E182–E186. DOI: `10.1503/cmaj.190334`. URL: `https://doi.org/10.1503/cmaj.190334`.

[18] Jonathan A. Edlow, Kiersten L. Gurley, and David E. Newman-Toker. "A New Diagnostic Approach to the Adult Patient with Acute Dizziness". In: *The Journal of Emergency Medicine* 54.4 (Apr. 2018), pp. 469–

483. DOI: `10.1016/j.jemermed.2017.12.024`. URL: `https://doi.org/10.1016/j.jemermed.2017.12.024`.

[19] Ottar Sjaastad et al. “Grading of headache intensity. A proposal”. In: *The Journal of Headache and Pain* 3.3 (Dec. 2002), pp. 117–127. DOI: `10.1007/s101940200029`. URL: `https://doi.org/10.1007/s101940200029`.

[20] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. *Redirected Walking*. 2001. DOI: `10.2312/EGS.20011036`. URL: `https://diglib.eg.org/handle/10.2312/egs20011036`.

[21] Niels Christian Nilsson et al. “15 Years of Research on Redirected Walking in Immersive Virtual Environments”. In: *IEEE Computer Graphics and Applications* 38.2 (2018), pp. 44–56. DOI: `10.1109/MCG.2018.111125628`.

[22] Mahdi Azmandian et al. “Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance.” In: *ICAT-EGVE*. 2015, pp. 93–100.

[23] Emil Kjenstad et al. “Experiencing Rotation and Curvature Gain for Redirected Walking in Virtual Reality”. In: *International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE ’23)*. Vancouver, BC, Canada: ACM, 2023. ISBN: 979-8-4007-0189-4/23/06. DOI: `10.1145/3592834.3592874`. URL: `https://doi.org/10.1145/3592834.3592874`.

[24] Timofey Grechkin et al. “Revisiting Detection Thresholds for Redirected Walking: Combining Translation and Curvature Gains”. In: *Proceedings of the ACM Symposium on Applied Perception*. SAP ’16. Anaheim, California: Association for Computing Machinery, 2016, pp. 113–120. ISBN: 9781450343831. DOI: `10.1145/2931002.2931018`. URL: `https://doi.org/10.1145/2931002.2931018`.

[25] Betsy Williams et al. “Exploring Large Virtual Environments with an HMD When Physical Space is Limited”. In: *Proceedings of the 4th Sym-*

posium on Applied Perception in Graphics and Visualization. APGV '07. Tubingen, Germany: Association for Computing Machinery, 2007, pp. 41–48. ISBN: 9781595936707. DOI: `10.1145/1272582.1272590`. URL: `https://doi.org/10.1145/1272582.1272590`.

[26] URL: `https://www.mn.uio.no/ifi/om/aktuelt/aktuelle-saker/2021/jentene-strommer-til-informatikkstudiene.html`.

[27] Stéphane Bouchard, Geneviève Robillard, and Patrice Renaud. *Simulator Sickness Questionnaire–French Revision.* 2007. DOI: `10.1037/t73606-000`. URL: `https://doi.org/10.1037/t73606-000`.

[28] Kay M. Stanney, Robert S. Kennedy, and Julie M. Drexler. "Cybersickness is Not Simulator Sickness". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 41.2 (Oct. 1997), pp. 1138–1142. DOI: `10.1177/107118139704100292`. URL: `https://doi.org/10.1177/107118139704100292`.

[29] Christian T. Neth et al. "Velocity-Dependent Dynamic Curvature Gain for Redirected Walking". In: *IEEE Transactions on Visualization and Computer Graphics* 18.7 (2012), pp. 1041–1052. DOI: `10.1109/TVCG.2011.275`.

[30] Debra Boeldt et al. "Using Virtual Reality Exposure Therapy to Enhance Treatment of Anxiety Disorders: Identifying Areas of Clinical Adoption and Potential Obstacles". In: *Frontiers in Psychiatry* 10 (Oct. 2019). DOI: `10.3389/fpsyt.2019.00773`. URL: `https://doi.org/10.3389/fpsyt.2019.00773`.