Redirected Walking in Virtual Reality with Distractors

Exploring Acceptability, Detection Thresholds, and Effectiveness

Halvor Kristian Ringsby

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Department of Informatics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Autumn 2023

Redirected Walking in Virtual Reality with Distractors

Exploring Acceptability, Detection Thresholds, and Effectiveness

Halvor Kristian Ringsby

Redirected Walking in Virtual Reality with Distractors

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

Printed: Reprosentralen, University of Oslo

Abstract

Virtual Reality (VR) has seen rapid advancements in the past decade, allowing for increasingly immersive experiences. However, locomotion, or movement in VR, remains a significant challenge, particularly when the virtual environment is larger than the user's physical space. Redirected Walking (RDW) enables users to explore large virtual environments, through natural walking, by actively steering the user away from walls and obstacles in their physical space. Redirected walking is, however, far from its ultimate goal of enabling infinite walking in VR through redirection that is imperceptible to the user. In this thesis, we present our exploration of redirected walking thresholds and the incorporation of distractors in VR to enhance RDW's effectiveness.

We developed a purpose-built system to test RDW, focusing on three redirection techniques (rotational gain, curvature gain, and translation gain). We also created two distractors to study the effect of Improved Redirection with Distractors on rotational gains. Two user studies were conducted to evaluate users' reactions to RDW and the influence of distractors on the experience.

Our findings show that user acceptability of RDW diminishes while the detection rate increases as the intensity of redirection increases. Results show that users are able to detect redirection of as low as 10% amplification for rotation gain and 7.5m radius for curvature gain. Thresholds for acceptability were identified between 40% and 55% amplification for rotational gain and a 5.0m radius for curvature. Though no specific threshold was found for translational gain, users seemed to not notice redirection until passing 25% amplification, and in terms of acceptability, even stronger gains appear feasible. Furthermore, distractors strongly influence the effectiveness of rotational gain, prompting increased head rotation from users by up to 43%.

In conclusion, this thesis broadens our understanding of RDW in VR by exploring the acceptability and detection thresholds for various redirection techniques and examining the impact of distractors on user experience and redirection efficacy.

Acknowledgments

I would like to thank my supervisor, Carsten Griwodz, for his excellent guidance, support, and encouragement throughout the writing of this thesis. I also extend my appreciation to Emil Kjenstad and Rahel Gaeb for serving as wonderful sparring partners in discussing thesis-related topics. Moreover, I thank all members of SIN-LAB for their keen interest and insightful suggestions related to the work presented in this thesis.

Finally, I would like to acknowledge my friends on the 9th floor of IFI for being great motivators and company throughout this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Extended Reality (XR), a term that includes virtual reality (VR), augmented reality (AR), and mixed reality (MR), is a field of technology that has seen rapid advancements in the past decade. XR technology is focused on providing the user with an immersive digital experience by showing them computer-generated environments or overlaying such elements onto the real world.

Virtual Reality (VR) allows users to experience and interact with fully digital environments as if they were in reality. This is possibly done by having the user wear a head-mounted display (HMD) which provides stereoscopic 3D vision into a computer-generated landscape. The user's movement is then mapped to the virtual experience to reflect their actions in the virtual environment. Thanks to technological advancements in the last decade, VR has become increasingly accessible and immersive, blurring the lines between virtual and real-world experiences.

However, despite growing adoption and significant progress in the technology behind VR, a large issue with achieving truly immersive experiences is the locomotion, or movement, in VR. If the virtual world is much larger than the physical space in which the user is present, how do we allow them to explore this virtual environment without crashing into walls or obstacles in the real world?

Due to this limitation, developers of VR experiences are often forced to decide how they want the experience to play out. For example, if the experience requires the user to explore a virtual environment larger than their Real Environment (RE), then an alternate locomotion scheme is required. Many schemes exist to facilitate this. However, to achieve this, they often decouple movement in VR from the real world. Although this method solves the issue of locomotion in VR it comes at the cost of immersion and naturalness in the experience by having a direct disconnect between what the user is doing, and what is being presented to them.

Although there are some possible solutions to this issue on the rise, such as VR treadmills, these are often inaccessible and unaffordable to most people. Therefore, to address the issue, we turn to Redirected Walking

(RDW) as a possible remedy. RDW is a method that can be applied to VR experiences to allow for extended or infinite walking in virtual environments when physical space is limited. First introduced in 2001 by Razzaque, Kohn and Whitton [34], this technology attempts to actively steer users so they do not leave their designated area. The steering can be done using various methods but commonly works by applying changes to the virtual environment or experience so that the user is forced to walk in a specific direction or pattern. Ideally, these adjustments are made imperceptibly to the user to maintain optimal immersion.

While RDW has seen gradual advancements since it was introduced in 2001, however, RDW should still be considered to be in its infancy as the use of RDW is seemingly contained in research applications and has not seen widespread adoption. Among the RDW research, there appear to exist contradictory results on the usability and effectiveness of RDW. Additionally, as VR technology continues to advance, much of the research that has been done is becoming more and more outdated as their VR systems no longer hold up to those of the modern day.

Furthermore, as more immersive and extensive VR experiences are becoming viable, there are many elements that could lead them well to RDW. One example of such is the use of distractors, usually included as enemies or non-player characters, in virtual environments that can be used to increase the effectiveness of RDW. Following the introduction of improved redirection with distractors (IRD) in 2009 the idea has produced promising results in extending the capabilities of RDW, but compared to RDW it has seen less research interest.

To this end, due to the lack of consistent findings, outdated studies, and lack of research into IRD, we note the need for a study that examines redirected walking in VR, as well as improved redirection with distractors.

1.2 Research questions

In this thesis, we wanted to focus on the use of Improved Redirection with Distractors (IRD) and the possible benefits this might bring to Redirected Walking (RDW). To this end, we also explore how users respond to general redirected walking as a means to calibrate our setup.

We, therefore, propose the following research questions for this thesis:

RQ $_1$ What are the thresholds for where users deem RDW unacceptable?

RQ $_2$ What are the detection thresholds for RDW with no distractors in the environment?

RQ $_3$ Does having distractors in the virtual environment affect the detection thresholds of RDW?

RQ $_4$ To what degree can having distractors present in the Virtual Environment (VE) improve RDW?

1.3 Scope and Contributions

Through this thesis, we hope to expand on the topic of Redirected Walking by exploring the detection thresholds and the level of acceptability users have towards various redirection techniques. Additionally, we place a strong focus on the use of distractors for redirected walking and aim to contribute to the relatively little research that has been done on distractors and uncover how they affect both the experience and the redirection.

This thesis will contribute to the research on redirected walking (RDW) by: (1) constructing a purpose-built system to test RDW, as there is a lack of updated systems available; (2) frame-focusing on fundamental aspects of RDW, employing a generalized controller with support for three subtle redirection techniques (rotational gain, curvature gain, translation gain) and a resetting controller as a safety measure; (3) creating two distractors to study the effect of IRD on rotational gains; and (4) designing and executing a user study to evaluate and gain insights into users' reactions to RDW in the developed system.

1.4 Thesis Structure

The structure of this thesis is organized into eight chapters, designed to address and explore the following key elements:

Chapter 2: Background In this chapter, we introduce ample background knowledge on virtual reality and redirected walking so that the reader is able to understand the topics in this thesis thoroughly.

Chapter 3: Methodology In this chapter, we detail the research method and approach that we employed for our thesis.

Chapter 4: Implementation In this chapter, we describe the implementational details of our purpose-built solution to RDW in Unreal Engine 5.

Chapter 5: Study Setup In this chapter, we describe how we set up the two user studies we conducted for this thesis.

Chapter 6: Results In this chapter, we review and analyze the results that were gathered during the user studies.

Chapter 7: Discussion In this chapter, we explore and interpret the results from Chapter 6.

Chapter 8: Conclusion In this chapter, we summarize the work done and key findings for this thesis and suggest future work.

Chapter 2

Background

In order to fully understand the benefits and limitations of Redirected Walking (RDW), it is first necessary to have a solid understanding of the underlying technology, Virtual Reality (VR), upon which it builds. VR is a complex and large ecosystem that involves a range of technical factors, as well as various aspects of physical and psychological aspects. As Redirected Walking (RDW) is simply a tool meant for enhancing the VR experience, fully grasping the potential impact of RDW without first having a thorough understanding of VR can prove difficult.

The goal of this chapter is to provide the reader with ample background knowledge on both VR and RDW so that the reasoning, goal, and subsequent results of this study are well understood.

2.1 A Brief Retrospective on VR Technology

VR has many beginnings depending on how you define it. If the requirements are that the user can see an alternate static landscape then one can argue that VR was introduced as early as the 1830s with the inventions of the stereoscope. This device allowed people to gaze upon still images in 3D by having images projected stereoscopically, meaning that each eye sees a slightly different picture [25]. However, if the definition requires that the user is able to experience an immersive scenario with moving pictures then one of the first VR devices would be the Sensorama by Morton Hellig in 1962, The Sensorama was able to project a stereoscopic film to the user while simultaneously stimulating other senses such as movement, touch, and smell [12]. Following Hellig's innovation within the field, the technology grew steadily with the first motion-tracked VR HMD, Headsight, being developed for military use in 1961. The Headsight was however not intended for virtual experiences, but rather for exploring dangerous real-life scenarios. The first idea of an introduction of a true VR-oriented HMD was put forward by Ivan Sutherland in 1965 and laid the foundation of VR [12].

In the wake of Sutherland's core principles of VR [12], the field was steadily moving forward with the technology that was available at the time, with the term *Virtual Reality* first being coined in 1987. While the technology was improving and VR became more and more advanced, one common attribute among all developments was that it was not consumer-oriented, which meant usage, affordability, and availability were low. Nearing the 1990s large actors such as SEGA and Nintendo were entering the field and attempting to create consumer-grade VR for gaming. However, possibly due to a lack of hardware capabilities, SEGA's HMD was never finished and Nintendo's attempt was quickly discontinued due to low profits [8, 12].

Following the failed attempts of the 1990s, VR technology would not see major development for consumer-grade hardware until VR's apparent resurgence around the year 2010. In the time after the *Oculus Rift* was prototyped, funded, and its ultimate release, the field of consumer VR technology started booming [2, 11]. With the Oculus and HTC Vive at the forefront, the large-scale adoption of VR was becoming more and more plausible. Ultimately, this boom resulted in VR becoming accessible to consumers due to lower costs, more powerful computer hardware, and large investments by big tech companies [12].

2.2 Current state of VR

VR technology has become increasingly widespread, and its hardware has become more readily available to consumers. As of April 2023, the VR PC hardware market is dominated by Oculus, with a market share of over 50% across two of their Head-Mounted Displays (HMD)s, while the remainder is split amongst various other manufacturers such as Valve and HTC *Steam Hardware & Software Survey* [36]. It is important to note that Oculus has been renamed to Meta following the renaming of Facebook; however, for the purposes of this thesis, the original name will be used. Similarly, as Oculus dominates the PC VR landscape, Sony is making significant strides in the console market with their PSVR 1 and PSVR 2 systems.

While the consumer market for VR is primarily oriented towards games and entertainment, the technology is also being utilized or researched for applications in fields such as education, healthcare, military, and workplace training [6]. In gaming, VR is mainly employed to immerse the user in a virtual world by minimizing ties to reality, thereby delivering a heightened gaming experience. Within fields such as healthcare, military, and workplace training, VR can be used to simulate scenarios that would otherwise be plagued with increased risk, cost, and limited reproducibility if attempted through traditional means. Additionally, research indicates that, in some cases, VR training can lead to higher performance than conventional methods [6, 9].

Despite the growing adoption and diverse applications of VR technology, there are still prominent limitations and challenges to be addressed in the current landscape. One major hurdle is locomotion in a limited physical space, which could constrain users from moving freely and naturally within the virtual environment [23]. Another challenge is cybersickness, also called VR sickness, a phenomenon akin to motion sickness, which occurs due to discrepancies between the user's visual cues and vestibular system [21]. Such discomfort can hinder the widespread acceptance and adoption of VR technology for various applications. Lastly, achieving a genuine sense of presence—the feeling of truly being within the virtual world—remains an ongoing challenge. As researchers and developers continue to overcome these limitations, the potential impact of VR in various fields is expected to increase, offering more immersive and effective experiences for users.

2.2.1 Locomotion in VR Experiences

One of the most significant challenges when it comes to VR experiences where the user is standing is that they must be restricted or adapted to allow for exploration of the full Virtual Environment (VE), within an Real Environment (RE) that is often significantly smaller. A limited RE is a large limitation for VR as it often constrains the user's ability to experience the VE. If the experience involves walking, running, or jumping, this is especially relevant as the user would need an alternative, usually less intuitive and natural, method to do these interactions as it might be dangerous, or simply impossible, in a small RE [23].

To allow users to freely move in the VE while being constrained by the size of their RE, developers must make use of clever locomotion control schemes. Generally, these methods allow the user to move in the VE without moving, or by moving in place, in the RE. Employing such a control scheme, instead of having the user walk naturally, makes VR more accessible for people who do not have a large RE. It also opens up the possibility for developers to create very large, or infinite, worlds, without having to worry about conforming to the size of an RE [2].

(a) Teleportation-based (b) Controller-based (c) VR Treadmill [14]

Figure 2.1: Various locomotion schemes for VR

- **Controller-based locomotion**: This method utilizes standard input systems like keyboards or handheld controllers, often with an attached joystick, for virtual movement. While easy to implement, it has significant drawbacks, such as motion sickness [2, 17, 47].

- **Teleport-based locomotion**: Users point at a location to instantaneously teleport there. While this non-continuous approach is popular, it may cause disorientation and is not suitable for experiences requiring continuous movement. Motion sickness is generally less prevalent in this method [2, 17]

- **Gesture-based locomotion**: This scheme allows for movement in the VE through physical movements or gestures in the RE. Examples include "walking-in-place" or leaning-based motion, offering an immersive, natural experience, often reducing motion sickness compared to controller-based methods [2, 17].
- **VR treadmills**: Physical installations where users are harnessed to a fixed position, enabling "walking-in-place" in the VE. This approach allows for free exploration while maintaining a high sense of presence [14, 45].
- **Room scale/Real walking**: A continuous, physical locomotion scheme with limited range in the VE. While restricted by the size of the RE, it provides accurate positioning, precise movement, and a natural walking experience [2].

2.2.2 Cybersickness in VR

A key issue that often accompanies VR usage is cybersickness. This problem, closely akin to motion sickness, is a major hindrance to the widespread adoption and usage of VR systems as a non-negligible fraction of VR users report symptoms of nausea, dizziness, and disorientation during or after VR sessions [21, 26]. In some cases, users report that the effects can linger for several hours before subsiding, and in extreme cases even as long as many days. Therefore, it can be argued that it is important to be aware of the possible reasons for cybersickness when working with VR to avoid causing unintentional discomfort for users.

Predicting the manifestation of cybersickness on an individual basis remains a complex task, due to numerous contributing factors – both technological and personal in nature. A number of theories have been proposed in attempts to explain the occurrence of cybersickness, LaViola [21] presents three of the most common; sensory conflict theory, postural instability theory, and poison. Each of these theoretical frameworks presents its own set of strengths and weaknesses, rendering it arduous to pinpoint a single prevailing cause for cybersickness. Nevertheless, these theories have helped the development of approaches designed to mitigate symptoms associated with cybersickness. However, despite such advancements, the effectiveness of these methodologies in all instances has not been fully substantiated.

- *Sensory Conflict Theory*: This theory is based on the premise that discrepancies between the senses which provide information about the body's orientation and motion cause a perceptual conflict that the body does not know how to handle. With cybersickness and motion sickness, the two primary senses that are involved are the vestibular sense and the visual sense. These sensory conflicts arise when the sensory information is not the stimulus that the subject expected based on their experience.

- *Postural instability theory:* This theory is centered on the idea that one of the primary behavioral goals in humans is to maintain postural stability in the environment. Postural instability will result whenever an animal links its control to patterns of stimulation that have ceased to be specific to those environmental conditions for which the control is appropriate. Therefore, this theory states that the cause of motion sickness and cybersickness is prolonged postural instability.
- *Poison theory*: This theory attempts to provide an explanation for why motion sickness and cybersickness occur from an evolutionary standpoint. The theory suggests that the ingestion of poison causes physiological effects involving the coordination of the visual, vestibular, and other sensory input systems. These physiological effects act as an early warning system that enhances survival by removing the contents of the stomach. The adverse stimulation found in some virtual environments can affect the visual and vestibular system in such a way that the body misreads the information and thinks it has ingested some type of toxic substance thus causing disturbing symptoms which lead to an emetic response.

Factors to Cybersickness in VR

There are various factors that can instigate cybersickness. These factors can be related to technology or to the individual. The work by LaViola [21] identified the following set of factors:

- Display and Technology Issues: Imperfections in technology, such as position tracking error, lag, and flicker can contribute to inducing cybersickness.
- Gender: Women appear to be more sensitive to cybersickness. A possible reason for this is that women generally have a wider field of view, which has been shown to correlate to the level of cybersickness [24].
- Age: Age differences play a factor in cybersickness susceptibility. Susceptibility is greatest between the ages of 2 and 12 years of age and decreases rapidly from 12 to 21 years and more slowly thereafter.
- Illness: Illness has been shown to be a contributing factor that increases a person's susceptibility to cybersickness.
- Position in the Simulator: Positioning the subject in the VE can also play a role in the individual's susceptibility to cybersickness. For example, sitting appears to be the better position in which to reduce cybersickness symptoms since it would reduce the demands on postural control.

- Non-Real-Locomotion: Using locomotion schemes in VR that do not stimulate any senses than the visual aspect, such as joystick movement, can contribute greatly to following the sensory conflict theory [2].

To combat cybersickness, several methods have been proposed to alleviate cybersickness in Virtual Reality environments. One such approach is the utilization of subtle dynamic field-of-view (FoV) modification (Figure 2.2). This method encompasses the dynamic alteration of the FoV within the virtual environment, contingent upon the user's movement. By lowering the FoV during swift head motions while augmenting it during less rapid movements, it becomes feasible to reduce the sensory discord that contributes to cybersickness [7].

Figure 2.2: Image illustrating the usage of dynamic FoV for cybersickness mitigation.
Image source: Fernandes and Feiner [7]

Another strategy to mitigate cybersickness entails implementing rest frames, which inject frames of the physical world in-between frames from the virtual world to ground the user in the physical space. This has the potential to diminish disorientation and sensory conflict, both known factors in causing cybersickness [21]. To mitigate the effect of sensory conflict in the realm of locomotion, having schemes that stimulate senses other than vision can help. Real walking, for example, will more or less eliminate the sensory conflict that takes place, optionally schemes that partly use a physical aspect may help limit the conflict [2]. In addition, offering users an adaptation program, designed to incrementally increase their exposure duration to the virtual environment, may facilitate their acclimatization and subsequently curtail symptoms associated with cybersickness [21].

2.3 Redirected Walking (RDW)

First proposed in 2001 by Razzaque, Kohn and Whitton [34], Redirected Walking RDW is a locomotion technique for room-scale, real walking, VR experiences that aims to enable extended, or infinite, walking in the VE with a constrained RE. With RDW the underlying system creates the illusion that the user can move freely in the VE, without being bound to the RE, by altering the user's perception of movement within the VE. The modifications done by the system can either be changes made to the user's view such as adding or amplifying rotation, or changes to the VE itself by changing the properties or architecture of the VE[29].

To allow for the best VR experience many solutions to RDW approach the problem with the hopes of being imperceptible. An imperceptible RDW system is one that is able to apply changes to the user's experience so subtly that users are not able to detect them. However, by aiming to be imperceptible the techniques are generally limited to how aggressive they can be, as any large changes will be easily picked up by the user. Moreover, by being limited to less aggressive methods the required size for the RE increases which might limit accessibility to RDW[23, 29].

Achieving redirection, be it imperceptible or not, can be done by employing various *redirection techniques*, in conjunction with a *redirection controller*. Redirection techniques serve as the backbone of any RDW system by describing practically how a user should be redirected. Controllers on the other hand generally have less say in how the redirection is applied, but rather monitor the state of the VR experience to determine which techniques should be used, and to where they should attempt to redirect the user [1, 13, 27].

The main objective of RDW is generally to prevent the user from colliding with physical boundaries or obstacles, however, this is not always possible. As the effectiveness of RDW can depend on how the user acts in the VE, in addition to how they respond to the redirection, the full completion of the objective can be hard. With some approaches to RDW the results can be that rather than completely eliminating breaches of the RE boundary, it is only able to decrease the amount. To this extent, RDW is often paired with a *resetting* mechanism that will upon a boundary breach, intercept and guide the user back to a safe space in the RE [13, 42, 43].

2.3.1 Redirection Techniques

A redirection technique can be defined as a single method of redirecting a user in a controlled fashion. Generally, these techniques apply some sort of modification or manipulation to the VR experience that a user is in so that the user must compensate with a movement with the opposite effect to counteract the modification [34, 39]. Some redirection techniques achieve redirection by exploiting *change blindness*, which is the fact that humans are sometimes unable to recognize changes in an environment that they are not familiar with [41]. Within the space of redirection techniques there exist many different methods, all either being slightly differing, or complete opposites. Although there exist many techniques for RDW, we will for the purpose of this thesis focus on *Rotational Gains*, *Curvature Gains*, and *Translational Gains*.

Taxonomy of Techniques

As there exist many techniques one can employ to redirect a user in VR, it is helpful to have a well-defined way of categorizing attributes and behaviors of the techniques. Suma Rosenberg et al. [42] proposes a generalized taxonomy for redirection techniques which sort techniques based on three main attributes:

1. *Geometric applicability.* Defines in which plane the redirection is applied. On one hand, you have *repositioning techniques* which are applied by manipulating the relationship between movement or positions in the RE and the VE, to where they are not always one-to-one. *Reorientation techniques* act in the rotational plane, where they modify the relation between rotation in the RE and VE.

2. *Noticeability to the user.* This attribute shows how the technique in question is meant to be perceived by the user who is being subjected to the redirection. *Overt* techniques are those that make no attempt at hiding from the user and oftentimes make it explicitly clear when redirection is happening. *Subtle* techniques however attempt to remain unnoticeable to the user so as to not draw any attention away from the VR experience.

3. *Content-specific implementation details.* Relates to how the redirection is being applied for a given technique. *Discrete* techniques are applied instantaneously with no integrated time component in the implementation. *Continuous* techniques, however, do have a time component and can either be applied over a set time interval or constantly applied throughout the VR experience.

Rotational Gains

Rotation gain is a technique for redirection that has had a wide range of studies examining it, in addition to being the technique proposed by Razzaque when he introduced RDW [34]. The technique can be classified as a continuous reorientation-based technique that can be both subtle and overt [13, 46]. Rotational gains can be defined as the ratio of which angular velocity[1] of the HMD in the RE is transferred to the user's virtual experience Θ ($\frac{\theta_{\text{Virtual}}}{\theta_{\text{Physical}}}$) [10]. The gain can be applied in any VR scenario as it only requires that the user performs rotations of the head (e.g., a user is around for an object or visually inspecting an environment).

By adjusting the rotational gain one can amplify or reduce the angular velocity applied to a user's view so that the user must physically compensate to achieve the intended amount of rotation in the VE. To reorient a user towards a specific point in the RE for example, a common approach is to amplify any rotation the user makes while turning *away* from the point and reduce the rotation when turning *towards* the point. The reasoning behind this is that if the user is turning their head to look at an element in the VE, which results in them looking away from the desired RE direction, then we want to limit the total needed rotation needed to look at the VE element. By amplifying the rotation applied in the VE we can turn what could have been a large rotation away from the goal into a *smaller* rotation. Inversely, if the user is turning their head towards the

---

[1] It is technically possible to use rotational gains for all axis of rotation (yaw, roll, and pitch), but yaw is the primary focus for most of RDW research [29].

desired direction, one can apply a reduced gain to force the user to exert *more* rotation in the RE to achieve their desired rotation in the VE [34].

Depending on the use case, rotational gains can be applied either subtly or overtly. Overt usage of the technique is mostly limited to resetting strategies, as extended use of aggressive rotational gains can be the cause of strong cybersickness in users. For general redirection where the rotational gains are applied continuously throughout the entire experience, the technique is mostly applied subtly by having the gains be low enough to where users do not recognize that any reduction or amplification is being made[29]. Some of the reasoning behind why rotational gains can remain imperceptible is the fact that it operate by modifying the effects of physical motion. This means that systems of the human body, such as the vestibular system, are receiving stimulation through the original movement which may help mitigate the user's ability to notice that redirection is being applied [38].

Curvature Gains

Curvature gain is another continuous reorientation technique that redirects the user by applying a constant rotation, with speed as a factor, to direct the user along a specified *curve* [38]. To have a user follow a set curve or an orbit, the technique will apply a rotation to the user's view, in the VE, in the direction that is opposite of the center of the orbit. The rotation is applied in the opposite direction so that the user sub-concisely compensates with a rotation in the RE that negates the added rotation. When the radius of the orbit is large enough, then the technique can function imperceptibly as the user will subconsciously correct their trajectory. [35, 38]. The technique can be applied in different ways, but in its general sense it will be based on the velocity, V, calculate how much rotation, $\Delta\theta$, is needed for a given time interval, ΔT to have the user follow an orbit, or circle, with the radius r: $\Delta\theta = \frac{360 V \Delta T}{2\pi r}$

Translation Gains

Translation gains is a relatively simple redirection technique, that can be classified as a continuous repositioning technique that can operate both subtly and overtly. The technique functions by modifying the relation between movement in the RE and movement in the VE, $G_{\text{Translation}} = \frac{\Delta\text{Pos}_{\text{Virtual}}}{\Delta\text{Pos}_{\text{Physical}}}$ [29, 47] By doing this the movement a user does in the RE can be either amplified or reduced based on the desired outcome. In its most simple state the technique can be applied to allow for extended walking by extending the traversable area in the VE, which is limited by the size of the RE, by a factor of $G_{\text{Translation}}$. Example-wise, if the user is using VR in a RE that is 2x2 meters, and $G_{\text{Translation}} = 1.5$, then the user would be able to traverse a VE with size 3x3 meters.

2.3.2 Detection Thresholds of Subtle RDW Techniques

To allow for the most immersion in VR while using RDW, subtle techniques are often employed. However, although such techniques may be explained as subtle, there is generally always a limit to how strongly the technique can be applied before the user notices that redirection is taking place [18]. The threshold where a subtle technique becomes overt can be measured in various ways depending on how the techniques are being applied [37]. In Razzaque's [34] first introduction of RDW, he tried to estimate thresholds but ultimately did not gain significant insight due to the lack of participants in the study. In the time since this attempt, there have been several other studies that have investigated the thresholds, however, it would seem that the results vary from study to study, as can be seen in figure 2.1. The reasoning behind the different results can possibly be due to a difference in testing methodology and system used for testing [10].

Gain	r_{real}[2]	Thresholds[3]	°/m [4]	Source
Translation	-	0.78 - 1.22	-	Steinicke et al. [37]
Translation	-	0.87 - 1.29	-	Bruder et al. [3]
Translation	-	0.86 - 1.26	-	Kruse et al. [16]
Rotation	-	0.67 - 1.24	-	Steinicke et al. [37]
Rotation	-	0.68 - 1.26	-	Bruder et al. [3]
Rotation	-	0.93 - 1.27	-	Paludan et al. [31]
Curvature	-	r >22.03m	3.5°/m	Steinicke et al. [37]
Curvature	-	r >14.92m	2.35°/m	Bruder et al. [3]
Curvature	-	r >11.61m	4.9°/m	Grechkin et al. [10]
Bending	1.25m	3.25	31.7°/m	Langbehn et al. [19]
Bending	2.5m	4.35	17.6°/m	Langbehn et al. [19]

Table 2.1: 'Detection Thresholds of Redirection Techniques' from Langbehn and Steinicke [18]'

[1] Radius of the, curved, walked path in the VE.

[2] For translation and rotation gains, the range of undetectable gains is stated. For bending gains, the maximal gain is stated. For curvature gains, the radius of the resulting arc in the real world is stated.

[3] For comparing curvature and bending gains, this notation is more suitable since it does not rely on the radius of the curves. It can be calculated if real and virtual radii are given.

2.3.3 Redirection Controllers

Generally, almost regardless of which technique is being used, the technique itself does not include logic that decides where a user is redirected, only the technical details of how they should be redirected. A *Redirection Controller*, sometimes also referred to as a *Steering algorithm* [1], is a key component that controls the various redirection techniques available. While a singular redirection technique might describe how to redirect, it is the controller's job to manage these techniques and tell them *where* to redirect [1, 29]. Redirection controllers work by continuously monitoring the different states of the VR experience, such as the user's location in the VE and RE, velocity, trajectory, etc. Through having situational awareness of the VR experience the controller decides where, and how, the user should be redirected. Some controllers may rely on a singular redirection technique to function, but it is also possible to have controllers that employ more than one [29].

Although many redirection controllers operate on the same basic principle of preventing the user from colliding with obstacles, there are several different approaches, each with its own nuances that set it apart from the rest. Of the range of controllers, the most prevalent methods are *Generalized Controllers*, *Predictive Controllers*, *Scripted Controllers*, and *Resetting Controllers* [29]. It should however be noted that while these controllers have elements that set them apart, it does not mean that one can not create a controller that includes elements from multiple classifications.

Generalized Controllers

Generalized controllers, also called *reactive* controllers, are a subset of controllers that will redirect the user based solely on the positional attributes in the RE. This means that it requires no further knowledge such as information about the VE, user intent, or similar. Compared to some other controllers this approach is quite novel, but this also comes with the benefit that it is quite flexible for use in any VR experience [29].

When Razzaque, Kohn and Whitton [34] first introduced RDW they proposed the use of three simple reactive controllers: (1) *Steer-to-center* (S2C): A controller that will constantly try to steer the user towards the center of the RE by applying continuous reorientation techniques. (2) *Steer-to-orbit* (S2O): A controller that guides the user along an orbit around the center of the RE. (3) *Steer-to-multiple-targets* (S2MT): An approach that will attempt to redirect the user towards one of several defined points in the RE. In a follow-up study on the three approaches, Hodgson and Bachmann [13] found that S2C will generally outperform the others. It is only in specific situations, walking straight forward in the VE, that S2O wins out over S2C.

In the time since Razzaque first introduced these algorithms, there have been various new additions to the list. An improvement that some of these additions introduce over the fairly simplistic algorithms introduced, is that they allow operation in RE's that have irregular shapes. An example of such an approach is the use of artificial-potential-fields (APF-RDW) [43], which will map the RE with weights that determine where the user should be redirected towards. Approaches like this work in irregular REs by not looking at where the center of the room is, but rather the distances to walls from a specific point. Additionally, these methods can also work as obstacle avoidance, as the "wall" can be inside of the RE. The use of this approach was seen to outperform S2C, buy at the cost of a more complex algorithm that must know the full extents of the RE [43].

Predictive Controllers

Predictive controllers are able to redirect users based on knowing what actions the user is most probable to do next [27]. By having information about what the future path of the user might look like in the VE, the controller can plan in a way such that the redirection best accommodates the path in the RE. Algorithms like FORCE will for example pre-plan several possible paths that the user can take within the VE, and calculate optimal way-points to steer the user towards using [48].

Although predictive controllers are generally able to plan redirection better and more dynamically than generalized controllers, they come at a cost of lowered flexibility. Whereas generalized approaches can function in more or less any scenario, prediction requires a deep insight into the layout of the VE in addition to sometimes having high pre-processing costs. For algorithms like FORCE, the VE must first be examined to determine the possible paths and optimal way-points [48]. Similarly, algorithms that predict based on previously observed user paths, such as neural-network approaches or LSTMs, require that there is a large amount of pre-recorded data that the algorithm can draw from [4, 27].

Scripted Controllers

Scripted controllers act in a very similar way to predictive controllers in that they have some knowledge about where the user will walk in the VE [29]. The difference between the two is that scripted controllers do not rely on advanced prediction algorithms to determine the most probable next action, but rather explicitly defined information by the developer [34, 41]. The advantage here is that the developer can carefully plan the path in the VE in such a way that it fits into the RE of which the VR experience takes place. Additionally, the developer can manually choose which redirection techniques to use at each point in the experience (e.g. curvature gains when walking down a straight path, and rotational gains when performing a task).

As with predictive controllers, this approach also sacrifices flexibility when compared to generalized approaches. By requiring the developer to always be aware of how a given VE, or path, should fit into the confines of the RE, it limits how much freedom the developer has when creating VE experiences [40] This approach can in some cases also be less flexible than predictive controllers as the scripted aspect must be baked in from the start of development, whereas predictive methods can adapt to new VEs given enough processing or data [4, 48].

Resetting Controller

Resetting controllers are a category of controllers that do not attempt to continuously redirect the user so that they avoid breaching the RE boundaries. Rather, resetting controllers will only activate when the user leaves the defined RE, and thus can not move further due to the risk of crashing into obstacles. If one compares it to the previous approaches which acted in a preventative outlook, this approach will act with an after-the-fact, last-line-of-defense mentality [23, 29].

As these controllers must generally act in response to a user breaching the boundaries, there is little time to apply redirection, and thus the most common approach is to use overt techniques. In addition to using overt redirection, many resetting controllers also prompt the user with some sort of visual feedback to notify them that they have breached the boundaries, and must be reset. Some common approaches include *2:1-turn*, which will enable an overt rotational gain and instruct the user to perform a 360-degree turn in the VE, which due to the rotational gains will only require a 180-degree turn in the RE. *Freeze-turn* will freeze the user's rotation and position in the VE and instruct the user to perform a 180-degree turn in the RE. *Walk-to-center* works by freezing the position of the user in the VE and instructing them to walk to the center before continuing the experience [23, 29, 43].

As briefly mentioned, continuous subtle RDW is not always successful in preventing a user from leaving the RE, and thus many approaches to the other controller categories also include a resetting controller as a fallback. As the use of this fallback is only in the cases where the main controller fails, the need for a *reset* is generally used as a metric for how good a RDW system functions. [29, 43]

2.3.4 Improved Redirection with Distractors (IRD)

Distractors as a component in RDW were first introduced by Peck, Fuchs and Whitton [32] in 2009 as a method to improve the effectiveness of RDW. In their research, they referred to the use of a distractor with RDW as *Improved Redirection with Distractors (IRD)*. A distractor is in its simplest form an element in the VE that has the aim of attracting the user's attention. Generally, distractors will either try and hide the fact that redirection is taking place by drawing the user's attention towards itself, rather than letting the user freely notice redirection. Distractors can also be used to induce rotation or movement from the user to allow redirection techniques more opportunities to redirect [37]. Following the first introduction of IRD Peck, Fuchs and Whitton [33] revisited the technique in 2010 with a more robust approach and was able to show that there are substantial potential benefits to using IRD.

Taxonomy of Distractors

When using IRD the developers must take several choices in the design of their distractors depending on which outcome they are after. Generally, distractors will all try to grab the user's attention by being a distracting element in the VE, however how they distract, and to which degree the user can interact with the distractor can vary. From the literature available there does not appear to be a formal taxonomy specifically developed for distractors, however, Nielsen et al. [28] has created a taxonomy for cinematic VR experiences that defines the various attributes of *cues* in VR cinematography. This taxonomy does however lend itself well to categorizing distractors, partly because research into IRD is a substantial part of the background [28]. Drawing inspiration from the taxonomy proposed by Suma Rosenberg et al. [42] this taxonomy also categorizes based on three main attributes:

1. *Explicit / Implicit:* A cue can be either explicit or implicit in how it diverts the attention of the user. Explicit cues are cues that either directly inform the user that they should shift their attention, or if the cue is something that is deserving of attention. Implicit distractors rely on the user noticing the cue without there being an explicit reason to divert attention. Explicit cues will generally cause voluntary shifts in attention, while implicit cues are more like to trigger subconscious shifts.
2. *Diegetics:* Relates to how the cue is incorporated into the VE. A cue that is part of the VE is said to be diegetic, while a cue that is external from the VE and only visible in the user's perspective is non-diegetic. A diegetic cue can be any element that is rooted in the VE and not an overlay or addition that has no connection to the VE (e.g. a monster, a signpost, etc.). A non-diegetic cue is a cue that is separate from the VE (e.g. a head-up-display in a game, a pop-up message, etc.)
3. *Interaction Freedom:* A cue can either allow free interaction with the VE while it is in effect, or it can limit the interaction the user is allowed to do. By limiting interaction freedom it is possible to more strongly redirect users by forcing the user to make certain actions, however limiting the freedom can also lower the sense of presence a user experiences.

In terms of RDW, most cues that have been explored in prior research have been explicit. Reasoning for why this is can be due to the nature of how implicit distractors can be overlooked by the user and therefore not generate an outcome that improves redirection. Moreover, some distractors, like the butterfly used in the study by Peck, Fuchs and Whitton [32], could be considered implicit, but as users have been instructed to pay attention to it, it should rather be thought of as explicit. Although outside of the scope of this thesis, one implicit distractor that has shown great promise for RDW is the use of faint blinking lights to allow redirection during saccadic eye movement [20].

Chapter 3

Methodology

This chapter guides the research design and procedure we employed for this thesis and how this relates to the research questions we have put forward. The following sections will first give an overview of the objective of this thesis, the systematic approach for designing our study, data gathering and analysis, and finally, the procedure used in our study.

3.1 Study Objective

This thesis investigates the application of redirected walking in virtual environments and its effectiveness, emphasizing user experience and thresholds of acceptability. The objectives of our investigation include examining the acceptability thresholds for users in RDW applications and exploring user responses to varying intensities of three redirection techniques: rotational, curvature, and translational gains. We aim to determine the levels of redirection that users find acceptable during VR experiences (RQ_1). In addition, we are interested in identifying the detection thresholds for each RDW technique without the presence of distractors. Also, pinpointing the precise thresholds at which redirection goes from subtle to overt. (RQ_2).

Our investigation also extends to the potential impact of distractors in the virtual environment, specifically analyzing whether the addition of distractors can influence the detection thresholds in rotational gains applications (RQ_3). Furthermore, we aim to evaluate the degree to which incorporating distractors can enhance the overall effectiveness of RDW. (RQ_4).

In summary, the primary objective of this thesis is to thoroughly explore the usability of RDW within virtual environments, including acceptability thresholds, detection thresholds, effectiveness, and the influence of distractors on these factors.

3.2 Study Method

In addressing the objectives posed in section 3.1, we have determined that conducting user studies is the most appropriate methodological approach. This decision is primarily grounded in our focus on investigating user responses and behavior, necessitating a methodology that facilitates direct examination of user interaction. Additionally, we note that user studies have been commonly employed in prior research (i.e. Chen and Fuchs [5], Langbehn et al. [19] and Razzaque, Kohn and Whitton [34]), thereby establishing their credibility as a reliable and effective approach for evaluating redirected walking.

Following this rationale, we first define the prerequisites necessary for our user study. We then explore a high-level design of the study, data collection methods, data analysis methods, and finally, the procedure for our study:

- **Testing Framework**: (Section 3.2.1)
- **Study Design**: (Section 3.2.2)
- **Data Collection**: (Section 3.2.3)
- **Data Analysis**: (Section 3.2.4)
- **Study Procedure**: (Section 3.3)

3.2.1 Testing Framework

We recognize that to test redirected walking, we must have a testing framework that can support the idea. The framework must provide all the required RDW techniques and intricacies needed to test and answer our research questions.

The framework for testing used during our studies is a self-implemented solution to Redirected Walking (RDW) that is integrated into Unreal Engine 5 (UE5). The technical details of our implementation are detailed in section 4.3. Our solution supports RDW in a room-scale VR setting, supporting the three techniques selected for this thesis, each with user variable parameters, a steer-to-center redirection controller, a resetting mechanism, and a custom boundary system for room-scale VR. The framework supports custom levels, distractors, and a rigid data collection suite.

3.2.2 Study Design

In designing our user study for this thesis we focus on three overarching ideas: (1) estimating detection thresholds for three redirection techniques, (2) effectively testing each redirection technique, and (3) examining the effects distractors might have on RDW in virtual environments.

In our thesis, we have chosen to split our study of RDW into two user studies, one focusing on RQ_1 and one which tackles $RQ_{2\text{-}4}$. In the following sections, we will detail how we realized the ideas listed for our two studies.

Additionally, to aid us in designing the studies we conducted a pilot study, that explored a subset of what the two main studies did. The full pilot study has been published in the *International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '23)* [15].

Estimation of Detection and Acceptability Thresholds

Answering $RQ_{2\text{-}3}$ requires that we examine the detection thresholds of RDW techniques, and in RQ_1, we must examine acceptability thresholds. To find a threshold where redirection techniques go from subtle to overt and from acceptable to unacceptable, we gather user responses to various intensities of redirection and compare the results. By comparing users' responses, we aim to spot a clear threshold where we find the highest possible intensity that users still can not detect or that they still find acceptable.

As we are testing for two different metrics, we know that users may still find redirection acceptable, even if it is overt. Thus, we must ensure that the chosen test cases (the different intensity levels) cover both aspects. In the next paragraph, we detail the method of which we used for choosing the given intensities for our user studies; a detail of how we aim to collect user responses that represent their response to the intensities can be found in section 3.2.3, while the method of analysis can be found in section 3.2.4.

Each of our redirection techniques bases its intensity on a parameter value. Therefore, we define a set of parameters for each technique that best covers the research goal for our user study. As a guide, we choose the parameter values primarily from previous research findings of detection thresholds (Table 2.1). We do this as we note that there have been somewhat dissimilar findings in prior research; thus, we do not blindly trust these findings. Therefore, with a starting point based on the findings, we deduce a set of parameters that cover both lower and higher intensities. The number of parameters and the granularity between them depends on the user study's scale. Additionally, we include a baseline test among all our parameters to assess the users' general response to the VE to explore any biases that might be present even without redirection. Finally, we recognize that the order of parameters may be a bias in itself, and thus, the order in which parameters are tested is randomized.

For the two user studies we conducted, the selection of parameters can be found in section 5.1.1 and section 5.2.1.

Evaluating Redirected Walking

To address RQ_1, where we investigate redirected walking without distractors, we design test scenarios (VEs) that aim to explore the redirection techniques reasonably and unbiasedly. We aim to explore how users can convincingly perceive if a given VR experience contains any redirection. Ideally, we want the users to move around a given Virtual Environment (VE) as naturally as possible to experience the redirection without irrelevant biases that would not be present in a practical application of RDW.

Therefore, the VEs that users experience are not empty and contain various elements to motivate natural walking. Had we left our VEs empty and bland, we might have risked the user over-concentrating their senses on detecting redirection due to the absence of other stimuli.

Additionally, as our redirection techniques have intricacies, we saw the need to design VEs that best fit the use of each one of them. The main goal here was to optimally test each technique by maximizing its use in a concise test. In our first user study, the VEs used to evaluate RDW are described in 5.1.2.

Evaluating the Effect of Distractors

For $RQ_{2\text{-}4}$, there is a strong focus on the effect of distractors on RDW. We accommodate the assessment of distractors in the VE by developing a test suite that reasonably tests this. For our study, we created a comparative method to test the users' responses to a VE where the presence of distractors is variable. By doing this, we can see users' behavior and responses while not being affected by distractors and note what changes upon introducing distractors.

In designing our distractors, we placed a lot of focus on having *diegetic* (Section 2.3.4) distractors in an immersive VE. For the distractors, we wanted to explore two main effects. First, we target a change in detection thresholds of redirection by designing a distractor for the VE that is always present in the VE. The aim was that the presence of the distractor would draw the users' attention so that less cognitive power remains to detect redirection. To draw the users' attention, we noted that there should be a *reason* the user should be aware of the distractor (e.g., posing a threat or interest to the user). Secondly, we introduce a distractor whose goal is to induce rotation of the user's head. This distractor's goal is to perform some action that engages the user to follow it with their gaze, and by simultaneously moving in the VE, will induce rotation of the user's head[1].

The two distractors employed in our user study are detailed in section 5.2.3, with their more specific implementation details explained in 4.3.4.

3.2.3 Data Collection

We opted to use a combination of qualitative and quantitative metrics as our research objectives involve examining user responses to redirected walking and the objective effectiveness of redirected walking. Concerning our research questions, we note that $RQ_{1\text{-}3}$ deals with users' subjective opinions, while RQ_4 draws the use of objective metrics that measure effectiveness. As multiple parameters are being tested per technique, gathering people's responses for each parameter is necessary. Therefore, our qualitative data is collected throughout the study, with users being asked for feedback immediately after a single test scenario. On the other hand, our quantitative data is collected during the test scenario itself by our

---

[1] As noted in the objective, the study of distractors will only consider rotational gains.

RDW system. Additionally, to uncover any hidden biases that might show themselves within specific demographics of users, we detail the collection of user-specific attributes that may have some effect on their VR experience.

The procedure for collecting qualitative data during the tests can be found in section 3.3, collection of user-specific attributes is found in section 3.3.3. An overview of the process behind gathering quantitative data is detailed in section 4.3.5. In this section, we aim to give an overview of the data we collect for our user studies, while an insight into the data analysis can be found in section 3.2.4.

Qualitative Data

In order to address research questions $RQ_{1\text{-}4}$, we gather a collection of qualitative data that can be used logically and reliably. The data we collect relates to a single redirection test scenario and explores the various points of interest in our research questions. In the following paragraphs, we explain the arguments for collecting various information before summarizing the questions we ask in tabel 3.1,

Noticability of Redirection. In RQ_2 and RQ_3, the answers to these are a direct analysis of when users start noticing redirection happening in the VR experience. Therefore, we query the user after each test scenario if they noticed any redirection during the test scenario.

Acceptability. For RQ_1, investigate the threshold where a redirection technique stops being acceptable to the user, a change in acceptability due to distractors can be related to RQ_4. We collected the users' opinions on if a given test scenario was acceptable. Due to possible difficulties with rating an experience as acceptable or not, we ask how acceptable the experience was on a scale.

Discomfort. In addition to the metrics drawn from direct links to the research questions, we also collect a secondary statistic to examine a side-effect of RDW. This side-effect is the cybersickness or general discomfort a user experiences during a test scenario. Although not directly attributable to a research question, this can provide insight into the acceptability ratings used for RQ_1 and RQ_4, as an increase in discomfort could likely lead to a decrease in acceptability. As with acceptability, we collect this metric as a point on a scale.

Category	**Answer Type**
Detection of Redirection	Yes / No
Acceptability	Scale of 1 to 5
Discomfort	Scale of 1 to 5

Table 3.1: Categories of qualitative data collected for test scenarios.

Quantitative Data.

During a given test scenario, we collected data so that we could analyze the behavior of the users, analyze the effectiveness of redirection, as well as being able to validate that our RDW system functioned properly. Ensuring the validity of our system is essential for all research questions. A malfunctioning system can lead to results that do not accurately reflect the answers we are seeking. Furthermore, the collection of data that can be used to analyze users' behavior and redirection effectiveness is directly related to RQ_4 in that it can subjectively tell us if there are any behavioral or effectiveness changes between test scenarios. Thus, we collected the following three types of data during test scenarios:

- *Physical User State*: Monitors elements in the Real Environment (RE), such as the position, rotation, velocity, and angular velocity of the HMD[2]. Used for analysis of user behavior.
- *Virtual User State*: Data in relation to the user's avatar in the VE. Monitors the same elements as done for the HMD, but in the VE. Used for analysis of user behavior.
- *Redirection State*: Records any data concerning the redirection happening within our system. This includes the gains being applied and redirection metrics. This data point is primarily used for the validation of the system.

User-Specific Attributes

We collected a wide range of user-specific attributes to uncover any potential biases within the participants of our user studies. As there was no sure way to deduce which demographics may hold biases for VR or RDW, we based our choice of attributes mainly on the findings of previous research into cybersickness and motion sickness. Additionally, we included some attributes based on proficiency in VR and video games as we recognized that these are only familiar to some.

To this end, we present the following user-specific attributes that we deemed relevant to exploring possible biases for VR and RDW:

- *Age*: There may be a cognitive difference between certain age groups. There may also be a difference in familiarity with VR.
- *Gender*: Research suggests that different genders may react differently to motion sickness. (Section 2.2.2)
- *Height*: Height may have some effect on cybersickness in VR experiences.
- *Medication against nausea*: This may affect the user's susceptibility to cybersickness as it is tightly linked to traditional motion sickness.

---

[2] We have no alternative tracking methods in our study, and thus the HMD is our only reference for the user's pose or location.

- *Vision Correction*: Wearing glasses/lenses may affect how comfortable it is wearing the HMD, as well as how well the user is able to see.
- *Sense of Balance*: People with a good sense of balance may be more likely to notice that they are being redirected.
- *VR Experience*: If the user is familiar with VR then they may be more likely to notice differences from their normal experience.
- *Video Game Experience*: People may be more aware of the VR experience, or more immersed if they play games regularly. It may also ease the speed at which they familiarize themselves with the experience, thus lowering confusion in early test scenarios.

3.2.4 Data Analysis

To draw any conclusions or results from our data, we define methods for analyzing it correctly and meaningfully. As we have decided to conduct two user studies investigating different aspects of RDW we define two focuses for our data analysis. Per the research questions and objective of the first study, this will focus on the qualitative aspect of user response within our test scenarios. For the second study, we also consider the qualitative data. However, as we are also trying to evaluate the potency of RDW with distractors present, we also analyze the quantitative data collected.

To best evaluate each of the techniques we have chosen and the effectiveness of distractors, we have developed the following structure for data analysis. In Figure 3.1, we have four main categories in the structure, each describing the purpose of the various data points.

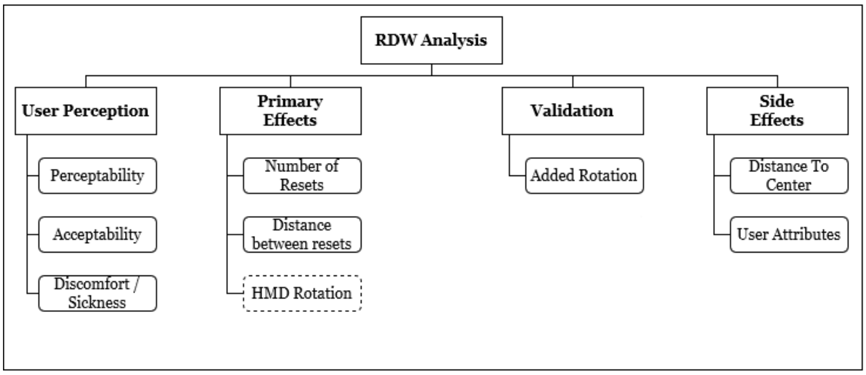


Figure 3.1: Diagram of data categories for analysis.
(Under Primary Effects, HMD Rotation is only applicable for user study 2.)

In our structure, *user perception* and *primary effects* will be used to find answers to our research questions. More specifically, perceptibility is used to answer RQ_1 and RQ_3, acceptability relates to RQ_2, and lastly, HMD rotation will help us answer RQ_4. Within primary effects, metrics are also used to measure the general benefit from our RDW techniques and system. The *Validation* category exists to validate that our RDW system functioned as intended during tests. Finally, the *Side Effects* category was added to investigate any effects that do not directly relate to our research questions but that could help us gain insight into the behavior of participants when exposed to our tests.

Analysis of Variance

When analyzing data, it is essential to know if any differences found are statistically significant to prevent conclusions from being drawn from non-confident sources. For this purpose, we will employ analysis of variance (ANOVA) in our thesis to ensure that we know whether or not there is confidence in our analysis. With ANOVA, we can test if we should reject the null hypothesis (H_0), which states that no statistical significance exists in a set of given observations. If we cannot reject H_0, then we can not conclude or interpret our data with certainty.

We will primarily use a one-way ANOVA to check if there is a statistically significant difference between various groupings of data. However, as ANOVA assumes that the same participants are present in all groups, it is impossible to use this for participant attributes (as a single person can not be both short and tall). Therefore we will employ the Kruskal-Wallis method for participant attributes.

User Study 1

As noted in section 3.2.3, our first user study focused on the general application of RDW. In the study, we aimed to answer two research questions, but to this end, we needed analysis methods that would allow this. First, for RQ_1, we need a method for determining detection thresholds for RDW. Secondly, for RQ_2, a method to find the acceptability thresholds of RDW is required. Additionally, as the validity of our RDW system is a crucial part of tests being valid, we need methods to analyze the system's behavior. Therefore, we defined two main methods for analysis that covers the research questions and one method to validate our system:

- *Detection Thresholds.* To determine where the threshold lies for detection for the participants, we analyzed their answers to the question of redirection detection. This question has a boolean answer that we convert to an integer value ($yes = 1$, $no = 0$). The values were then run through ANOVA to determine if we could reject H_0. If we can do this, we inspect the means to determine the subtle and overt redirection intensities. We chose an upper limit where if 10% or more of participants detected the redirection would be considered overt. The 10% threshold was chosen to investigate the average user's opinion on redirection, and therefore we added a buffer to offset any potential biases in specific demographics or users.

- *Acceptability Thresholds.* The threshold for acceptability was evaluated based on the participant's score on the acceptability scale. The answer to this question is an integer value between one and five. If we can reject H_0, we inspect the mean values to determine which intensities were acceptable and which were not. For acceptability, we set a floor of 4.0 when we consider a given parameter unacceptable. The reason why 4.0 was chosen was that it was labeled as *acceptable* when questioning, whereas 3.0 was labeled as OK.

- *Validation.* We inspected the metric *added rotation*[3] to validate that the RDW system was functioning as intended during a test. By looking at this variable, we could deduce if a set of test scenarios differed in the amount of redirection applied. The expected differences between intensities would be close to the proportional increase in intensity; however, some differences were expected due to the intricacies of the RDW controller.

User Study 2

Our second user study focused on how users responded to distractors within the VE. In RQ_3, we set out to answer if distractors could affect the detectability of rotational gains, which means we need a way to extract this from our data. Furthermore, in RQ_4, our goal was to investigate if distractors could hold any effectivity benefits to rotational gains. Therefore, we developed the following methods for analyzing our data:

- *Distractor Effect on Detection Thresholds*[4]. Using the same method for uncovering detection thresholds, we first find the thresholds of test scenarios with and without distractors. We then compared test scenarios with and without distractors to each other, grouped by the redirection intensities. If ANOVA states that we can reject H_0, we can compare how distractors affected the detection thresholds for a given intensity.
- *Overall Effect of Distractors.* The effect of distractors across the span of the entire test scenario was measured by taking the *average HMD rotation*[5] exerted by a user during the tests with distractors, and without distractors, and comparing them. The two cases were then put through ANOVA to ensure we could reject H_0.

3.3 Study Procedure

This section outlines the procedure we used when conducting both of our user studies. The goal of creating a well-defined procedure was to maintain a systematic and structured approach for both our user studies, as well as for each participant within the studies. The following is a list of key steps that were taken during our user studies:

- *Participant Recruitment* (Section 3.3.1)

---

[3] For translational gains, we will be looking at the distance walked in the RE vs. distance in the VE, as this technique does not deal with rotation

[4] We use the same method for finding differences in acceptability rating.

[5] As stated in the study design, our goal of the distractors was to increase HMD rotation

- *Physical Testing Location* (Section 3.3.2)
- *Pre-Study Questionnaire* (Section 3.3.3)
- *Experimental Task Introduction* (Section 3.3.4)
- *Experiment Execution and Live Data Collection* (Section 3.3.5)

3.3.1 Participant Recruitment

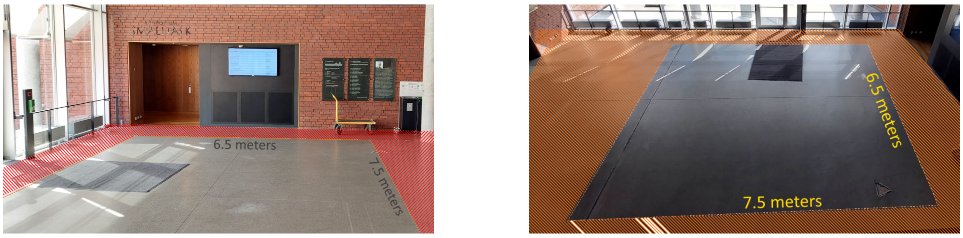
Overall we aimed to have a diverse group of participants. However, as we anticipated that it could be difficult to recruit participants, we did not have any selection process for our participants, and time slots were allotted on a first-come basis.

To recruit participants for our user studies we used a combination of several different methods. Firstly, we created a small website with information about our research in addition to a button that would take you to a sign-up page. On the page, one could then sign up for a specific time slot for their test. This website was spread around using posters at the Institute of Informatics at the University of Oslo. Secondly, participants were recruited from friends and fellow students. Lastly, our tests were conducted in a high-traffic, open space (See section 3.3.2) which allowed us to advertise to onlookers who then had the chance to sign up for a test.

3.3.2 Physical Testing Location

For redirection to be remotely applicable we identify the need to pick a suitable physical location that can support redirected walking. The location should have a wide, flat, and open space that is clear of any obstructions.

In our studies, the physical location, the RE, would be an open space in Ole Johan Dahls Hus at the University of Oslo. The area can be seen in figure 3.2, where the outlines are clearly shown. During the tests, we were able to use an RE with a size of 6.5x7.5 meters. To minimize the disturbance from the surrounding environment, we surrounded the RE with a plastic boundary band. As the location is inside an active university building, it incidentally lead to a great opportunity for recruiting new participants (See section 3.3.1).



(a) Side view of the RE

(b) Birds-eye-view of the RE

Figure 3.2: The RE used for user studies. The red area signifies the out-of-bounds areas. The numbers denote the lengths of the sides of the rectangular RE.

3.3.3 Pre-Study Questionnaire

Based on the points laid forth in section 3.2.3 we collect general information from participants prior to their testing. The attributes were gathered as part of the online schema used to sign up for the user studies. Table 3.2 shows the attributes in which we collected, and in what format it was collected.

Description	Type
Age	years
Gender	male/female/other
Height	centimeters
Medication against nausea	yes/no
Vision Correction	yes/no
Sense of Balance	1-5
VR Experience	1-5
Video Game Experience	1-5

Table 3.2: User attributes collected for user studies

3.3.4 Experimental Task Introduction

Before participants in our study are thrust into the various VEs, we thought it smart to first give them general information in advance to prepare them for the tests. Our reasoning behind this was that if participants became confused about the objectives within each VE it could reduce the quality of their experience, and possibly extend the length of tests by having to explain underway. To facilitate a simple method of providing this information, we supplied the participant with a pamphlet (See Appendix A) containing information that was related to the user studies.

In the pamphlet, there were two main elements. First, we give a brief word on safety during the study that states: A participant should for their own safety not exceed walking speed and that if they should at any point feel ill they may choose to terminate the test. Secondly, it gives an overview of the various VEs that the participant will encounter during the study, and it explains the tasks that the user must complete in the VEs. We chose to not include specific descriptions of the techniques as having the user be aware of *how* redirection was being applied could have biased their senses to be more focused on detecting the specifics of the redirection.

Furthermore, after a participant has put on the HMD, we first allow them to experience a *tutorial* VE. The VE allows the participant to acclimate to VR and become familiar with key aspects, the VE also introduces aspects that are relevant to the test scenario VEs of the user studies (e.i. picking up objects, resetting, and distractors). The tutorial VE was introduced mainly to have the participant able to experience standard VR for a brief period before the test as giving the users time to adapt to VR has been shown to reduce cybersickness (Section 2.2.2). Additionally, this VE serves as a place test conductors are able to verify that the various components of the RDW system are functional before starting the tests.

3.3.5 Experiment Execution and Live Data Collection

Once the participant has completed the tutorial, they were transported to the first test in the study, and they may then complete the task for the given VE. While the participant is in a live test, the test conductors do not provide any feedback or instruction, unless the user specifically requests help with the tasks[6]. We choose to do this because we foresaw it to be hard to control the information supplied during verbal communication, and that we could not ensure that this information would be consistent across participants, which then could lead to biases based on the information given. Additionally, having the participants concentrate on outside stimuli could deteriorate their VR experience.

Once a user has completed the task within a VE they are transported to an *intermission* VE that we designed to serve as a neutral area without redirection. In this VE the test conductors query the users for the qualitative data that is listed in section 3.2.3. Additionally, this level serves as a streamlined way for us to enact our relative actor placement strategy (Section 4.3.5). Finally, once all tests of the study have been completed by a participant, they are transported to a *credits* VE which shows the contributors to this RDW study, as well as informs the participants that they can remove the HMD. Following this, the participant has completed the study.

---

[6] Test conductors would in case of safety concerns (i.e. participant walks too fast, or if there are people inside the RE) relay this information to the participants.

Chapter 4

Tools and Implementation

In this chapter, we explain the tangible solutions to the various requirements and ideas that were defined in chapter 3. In section 4.1, we detail the reasoning behind our choices for the hardware and tools used for our research. In section 4.2, we will briefly explain how we went forwards in implementing our RDW system. Finally, in section 4.3, we will detail the actual implementation of our system and any choices made for the implementation.

4.1 Hardware

We employed two key hardware components during our tests, the HMD and the computer. In this section, we will show our arguments for our choices in their selection. The hardware was chosen to best fit our needs, but it should be noted that we were partly limited in the number of choices that we had available. Thus, the final choices may not represent the optimal choice but rather the optimal one among that available to us.

4.1.1 Choice of HMD

During the preliminary stages of our research, we experimented with various HMDs. We used an Oculus Rift early in development, which was tethered to our computer with a wire. This tether proved a problem, limiting the physical space we could traverse. Therefore, we explored solutions that would allow for a larger traversable area. Using a tethered HMD with a long cable was considered but ultimately not chosen due to how the cable could be a concern to the user, which could affect their experience. Within wireless HMDs, we considered the HTC Vive with its wireless module. However, the Oculus Quest and AirLink ultimately ended up being used.

The Oculus Quest is designed primarily as a closed system, where games are run locally on the headset. This setup reduces latency for HMD input processing and video responsiveness; however, we opted to use the Quest with *Airlink* for the most straightforward integration with Unreal Engine 5 (UE5). Airlink functions as Oculusẃireless VR solution by transmitting video from the host computer to the HMD and transmitting HMD input to the computer running the application. For our trails, we used a 5GHz hotspot to serve the Airlink, as attempts with simple 2.4GHz connections proved to be too weak. The Airlink setup was observed to be highly sensitive to obstructions between the WiFi hotspot and HMD, requiring a clear path between the HMD and hotspot, with a max range of approximately 10-15 meters in some cases.

Another notable aspect of using the Oculus Quest is its self-contained HMD tracking system, which eliminates the need for external tracking stations, such as those utilized by the Vive. Although this feature generally served as an advantage, allowing for an effortless transition between testing areas without extensive setup, the inability to use external tracking points, such as "Vive tracking pucks," was considered suboptimal and proved somewhat limiting to our implementation. However, it did not severely hamper our efforts.

4.1.2 Computer

In our research, we aimed to provide a seamless VR experience without being hindered by the computer's processing power limitations. To this end, we utilized a desktop computer with specifications that significantly surpassed the minimum requirements. Initially, we attempted to use a less powerful laptop and an older model of Intel NUC. However, both options proved inadequate in delivering the desired VR experience, ultimately prompting us to opt for a more powerful system. The computer we utilized is outfitted with an NVidia RTX 4080 Graphics card and a Ryzen 9 7800X CPU, ensuring ample processing capabilities and a smoother VR performance.

Despite the high-end hardware, we experienced occasional issues during our study, particularly with the GPU crashing randomly. This setback made our tests quite challenging, as we were often forced to restart multiple times during a single test. Although we could not pinpoint the exact cause of the problem, we implemented a few measures to mitigate the issue. By updating the GPU driver and underclocking the graphics card, we significantly reduced the frequency of crashes, ensuring that they became a rare occurrence and did not impact the overall results of our user studies.

4.2 RDW System

We must have a system that can support all our requirements and expectations to test the various techniques and methods we targeted for our user studies. To this end, we have looked at several solutions to satisfy the requirements. In our exploration, we searched for previous solutions to our needs. However, we ultimately found that what was available was either poorly documented, outdated, or overly complicated[1]. Thus, we concluded that we would implement our own version to ensure an RDW system that met our needs. Technical details on the implementation can be found in section 4.3.

4.2.1 Implementation Strategy

In order to create compelling VR experiences without reinventing the wheel, our project employed widely-used development tools. Since the most prevalent applications in VR are games, we opted to develop our project within a game engine that offers inherent VR support. We considered two main game engines—Unreal Engine 5 and Unity—as these are the most accessible engines, with extensive documentation and guides. As both engines are quite popular, support VR, and are user-friendly, our choice came down to their potential for modification. Although Unity can sometimes be seen as more straightforward for beginners, its closed-source nature may have hindered our ability to change core functionalities should we need to. In contrast, Unreal Engine is open-source, allowing us to change any detail of the engine's behavior, providing us with greater flexibility and adaptability. As none of the contributing members for this project had prior knowledge or skills for any of the two engines, we opted to use Unreal Engine 5[2].

During development, we quickly noted the modular nature of UE5, where all elements in the game engine are made with an object-oriented methodology. Thus, following this, we sought to implement our system modularly so that our solutions could be as flexible as possible. Additionally, due to the sheer size of UE5's source code, we found it quite time-consuming to explore and understand how we could change the source code to alter standard behavior. Therefore, we opted to avoid changing the source code directly wherever possible and instead implement extensions on top of common UE5 elements.

4.3 Implementation

The following section will cover details on how we implemented our custom RDW system within UE5. To minimize the time spent working on implementing traditional VR mechanisms and ideas, we base our system on the *VR Template* that Epic Games provide. This template offers seamless support for all mainstream HMDs and a significant level of modifiability.

---

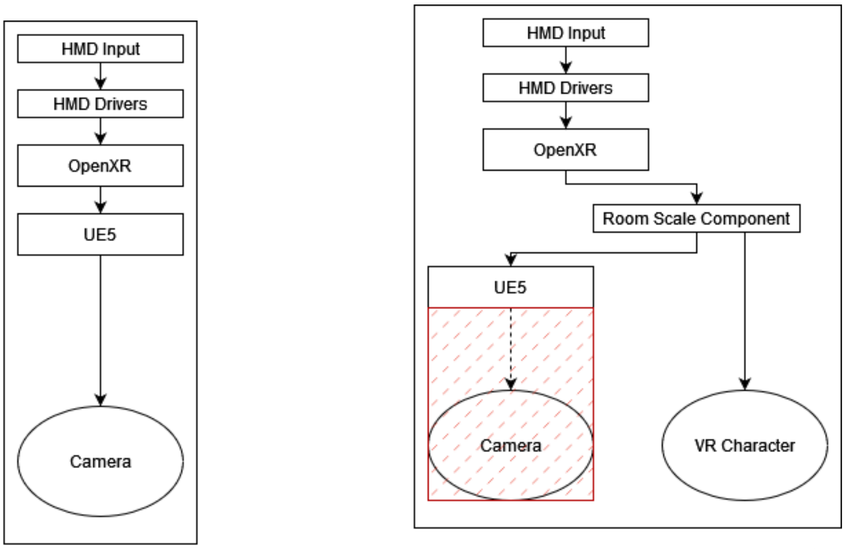
[1] From what we found, only two solutions were available at the time of writing this thesis. These were RDW Toolkit[44] and a fork of this called OpenRDW[22]. RDW Toolkit has been abandoned. While OpenRDW is still active but seems to not support the HMDs we had available, as well as seeming complicated to use. This solution was also found after we had started working on our own solution.

[2] At the time of development start, we used the Unreal Engine 5 preview, as UE5 was not fully released.

4.3.1 Preparing for Redirection: Modifying HMD Control Input

Initial attempts at applying a rotation to the VR character included in the template proved to be more complicated than we had hoped due to the implementation of how HMD input is applied to the virtual character. Any HMD control input is applied as a relative offset to the camera attached to the character, not the character itself. Attempts at a rotation of the character would cause the camera to swing around in orbit around the character, and any rotation applied to the camera would be ignored. Therefore, to rotate the camera's view, a transform must be calculated to apply an offset rotation and offset location to the character so that it causes the character to orbit the camera instead. We felt this transform to be an unnecessarily complex method of rotating the user's viewpoint. Therefore, we set out to simplify the process.

Our solution to the problem was to change how the HMD input is applied to the character by modifying and extending the framework UE5 uses for VR integration, *OpenXR*. OpenXR is an open-source framework for simplifying the integration of HMDs into applications by abstracting device-specific API use into a single shared API [30]. In UE5, OpenXR is integrated as a *plugin* and not a native part of the source code. As it is a plugin, we could extend OpenXR in a way that does not change the source code of UE5. In figure 4.1a, we show where OpenXR operates within UE5 and the VR template.



(a) Standard application of HMD input in UE5

(b) Our solution to HMD input. The red section signifies that no offset is being applied as it is a zero-vector.

Figure 4.1: Simplified illustration of how HMD input is applied within UE5. Lines represent the path that the input follows, from the physical HMD, until it reaches either the camera or VR character.

To fix our issue, locked the camera of the VR character directly to the character so that there no longer was any offset, thus allowing us to apply a rotation to the camera directly through the character. We ultimately modified OpenXR's `GetCurrentPose()` function to do this. This function is OpenXR's way of providing the current location and rotation of the HMD to Unreal Engine. This appeared to be a good approach if we wanted to change how UE5 interpreted HMD input.

Our extension to OpenXR involves creating a component called the Room Scale Component. When this component is active, it will register a callback to itself within the `GetCurrentPose()` function so that it can intercept any HMD data and modify it. To prevent the default behavior of offsetting the camera, we modify the reported HMD state so that UE5 believes that the HMD has not moved. This change stops any offset from being applied, and the camera stays locked to the character. Then, to ensure that HMD movement is still represented, the room scale component will apply the movement to the character directly (Figure 4.1b). Our solution, which attempted to be as noninvasive as possible while remaining efficient, may be perceived as a hack. But considering the alternative options, we believe the problem was solved efficiently. Overall we found that this solution worked well for our needs and provided greater flexibility and control when creating our VR experience.

4.3.2 Redirection Controller

We implemented a steer-to-center oriented redirection controller supporting two reorientation techniques, curvature and rotation gains. We also included support for translation gains, which is static and therefore does not follow the same logic as the two reorientation techniques. Our controller is implemented as a component that can be added to the VR character. The component offers a wide range of customizations for each technique implemented. Additionally, we implemented our resetting controller in the same component as they would have overlapping logic if separated.

For every frame rendered, also called a *tick*, the controller will perform a set of actions to decide how redirection should be applied. Since we are only testing a single technique at a time, the controller will only try to use a single technique at once.

Curvature gains

In order to implement curvature gains in our redirection controller, we guide the user along a circular path with radius r centered in the middle of the physical space. To steer along a circular path, we compute the appropriate rotation to be added for each frame. Given the radius of the desired curved path and the current velocity of the user V, we can estimate the time t required to traverse the entire circumference of the circle as $t = \frac{2\pi r}{V}$. By knowing the duration of a single frame ΔT, we can calculate the number of frames it would take to complete the circular path: $N_{\text{Frames}} = \frac{t}{\Delta T}$. We can then determine the necessary degrees per frame by dividing 360° by the number of frames it would take. These calculations culminate in Equation 4.1.

Furthermore, it is crucial to consider the position of the center of the physical space relative to the user (see Section 4.3.2). If the user has the center point on their right, the character is rotated to the left, and vice versa. During development, we noted that in some instances, a switch between being rotated left and right can cause a sudden jolt that is very noticeable to the user. Therefore, we apply *ramping* of gains to ease the transition (see Section 4.3.2). To further limit the oscillation between gains, we also included a *dead zone*, of 5°, for when the user faces directly towards or away from the center.

$$C_{\text{AddedRotation}} = \frac{360V\Delta T}{2\pi r} \tag{4.1}$$

Rotational Gains

Implementing rotational gains in our redirection controller necessitates computing the amount of rotation to be applied to the avatar for each frame. This computation is relatively simple, only requiring the multiplication of the gain parameter (G_{Rot}) with the change in yaw (ΔYaw), as demonstrated in Equation 4.2. Similarly to curvature gains, it is necessary to determine the center point of the room in relation to the user's position (see Section 4.3.2). However, to find which gain to use, *upper* or *lower* gain, we must also know the direction of rotation, ΔYaw. If the user is turning their head towards the center, we apply the lower gain, and if they are rotating away from the center, the upper gain is used. Additionally, as with curvature gains, we apply ramping (Section 4.3.2) when switching between gains.

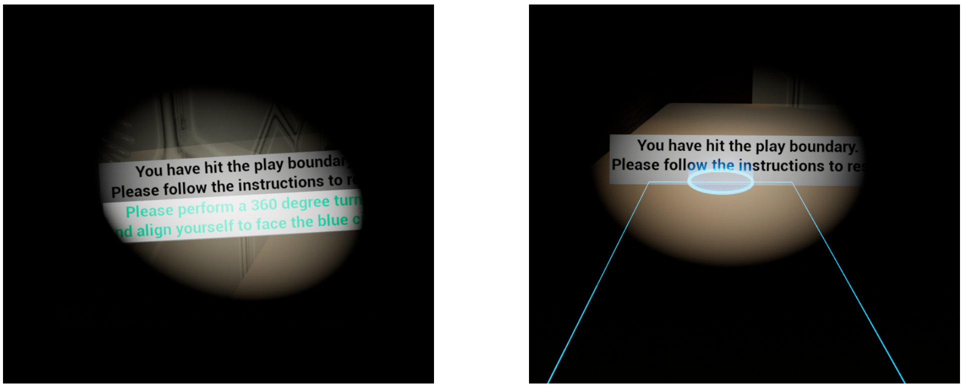
$$R_{\text{AddedRotation}} = \Delta\text{Yaw} \cdot G_{\text{Rot}} \tag{4.2}$$

Resetting

Resetting is an important mechanism in our application to prevent users from colliding with obstacles or leaving their designated play area. Several methods exist for resetting a user when they go out of bounds; however, we opted to implement two approaches that we believed were most effective.

- Point-To-Center: Requires the user to perform a turn in the VE so that they end up looking at the center of the RE. During this maneuver, a rotational gain is applied to the actor, causing them to complete a full 360-degree upon facing the RE's center. This method is also known as a 2:1 turn since the actor's rotation is close to double the user's 180-degree turn.
- Walk-To-Center: This method instructs the user to walk towards the center of the play area, with the reset considered complete once they reach the middle.

In addition to these two primary methods, our final resetting implementation incorporated various supplementary features:



You have hit the play boundar
Please follow the instructions to r
Please perform a 360 degree turn
nd align yourself to face the blue c

You have hit the play boundary.
Please follow the instructions to re

(a) Pop-up message for Turn-To-Center reset

(b) Path to the middle for Walk-To-Center reset

Figure 4.2: Example of a reset event with tunnel-vision active

- Tunnel Vision: Early in the testing process, we realized that the high gain and locked actor movement during resets made the original methods incredibly nauseating for users. To alleviate this issue, we added an effect to the HMD that narrows the user's field of view by introducing a black gradient, which obscures everything but the central portion of the screen. This method is inspired by Fernandes and Feiner [7].

- Overlay: To help users orient themselves during a reset, we added overlays that display the center of the RE, allowing the user to follow the overlay and align themselves more easily. Another overlay exists for the *Walk-To-Center* method, illustrating a path toward the center for the user to follow. The two overlays also act as a reference to the physical world, serving a similar purpose to rest frames (Section 2.2.2)

- Walk-Some-To-Center: Walking to the room's center is unnecessary in some scenarios, so we included a secondary option that prompts the user to walk a fixed distance toward the center. This method is well-suited for large rooms, as continually walking to the center can be tiring for the user.

- Combination: More than a single resetting method is often required. For example, if the user breaches the boundary and undergoes a Rotate-To-Center reset, they may still be outside the play area despite facing the center. In cases like these, a secondary Walk-To-Center is initiated.

- Aim-To-Orbit: Since curvature gains attempt to guide a user towards the center point along an orbit path, pointing the user directly at the center during a reset is suboptimal. This issue is further highlighted by the algorithm's *dead zone* around the center. To address this, we included a method that directs the user toward a point slightly offset from the center, placing them in orbit around the central point.

Finding the Center of the RE

In both techniques, we calculate the gain based on the user's pose relative to the central point of the environment. This pose can be determined by utilizing the vector cross product between the user's rotation (θ_{Real}) in the environment and the user's position from the center of the environment (r_{Real}), as illustrated in Equation 4.3. These vectors are represented in a 2D plane since our scenarios' third dimension (height) remains constant. If the result produces a positive number, the center of the environment is to the user's right; conversely, if the result is negative, it is to the user's left.

$$\text{Reference} = \theta_{\text{Real}} \times r_{\text{Real}} \tag{4.3}$$

Ramping of Gains

During the early stages of development, it was observed that relying solely on the mathematical descriptions of redirection techniques could lead to oscillations when the center point of the RE changed sides (left or right). Therefore, a *smoothing* element was introduced to facilitate seamless transitions between the gains applied.

The ramping mechanism is incorporated independently for both techniques. Regarding curvature gain, the ramping mechanism smooths the transition by adjusting the desired curve's radius. When changing the direction of rotation, the gain is initially increased to a high value (50m radius) before gradually decreasing to the desired value over 0.75 seconds.

In the context of rotation gain, a simple step algorithm was implemented to increase or decrease the gain by a predetermined amount per frame for one second. The simplicity of ramping rotational gains comes from the fact that the gains used will naturally pass the neutral state without other mechanics (e.g., from 0.9 to 1.3 will pass the neutral 1.0 gain). One caveat of ramping rotational gains is that if the user changes the direction in which they are rotating, but the center stays on the same side, then we do not perform ramping but rather an instant change. If we kept the standard ramping in this scenario, the gain would lag, causing the wrong gain to be applied.

4.3.3 Boundary System

A challenge encountered while exploring larger REs was the inadequacy of the built-in boundary system provided by our Oculus device. Specifically, it did not always permit the definition of the entire RE as a valid play area. To address this issue, we developed an alternative boundary system allowing us to define arbitrarily large REs.

Our boundary system is based on four user-defined corners of a quadrilateral within the RE. This quadrilateral defines the limits of the current play area. Subsequently, the component persistently checks whether the user is inside or outside the designated play area. A reset is triggered if the system detects the user breaching the boundaries. Although initiated by the boundary system, the actual implementation and handling of the reset are executed within the redirection component (See Section 4.3.2).

4.3.4 Distractors

To test the effectiveness of IRD we implemented two distractors into our VEs. Arguments behind the design of the distractors can be found in section 5.2.3. The following will describe the technical implementation of the gameplay mechanics included in the VE, in addition to the distractors themselves.

Gameplay mechanics

Within the VE that we designed for IRD testing (Section 5.2.3), we also added three additional gameplay mechanics to raise the users' immersion factor:

- *Chests and Gold.* Scattered throughout the VE, the user can find chests that are filled with gold. The chests include logic that causes them to first open upon touching the lock attached to it. Then, upon touching the gold inside, an animation of a *gold explosion* will be played. Once the user has touched the gold, they are awarded 50 points[3].
- *Info-menu.* If the user wants to check the remaining time, chests, and current point score, they may open a menu on their right hand by pressing the *A*-button on their joystick.
- *Projectiles.* Distractors may throw projectiles at the user during the test. The projectiles were implemented using UE5's *Projectile component*, which allows us to launch a sphere at the user by simply spawning it with a velocity vector in the user's direction. The projectile has a *lava* texture and a particle emitter that will spawn a shower of sparks. If the projectile collides with the VR character, it will play an explosion effect and de-spawn. It will also deduct 25 points from the total score upon impact with the character. If the projectile fails to hit the character or wizard, it will bounce along like a ball. After approximately 5 seconds, it will self-destruct and explode.
- *Blocking.* Affixed to the user's right joystick, in the VE, there is a shield. The projectile recognizes the shield as a special case. If the user successfully blocks a projectile with the shield, it bounces off. If the projectile was thrown by the wizard, then the projectile will be bounced directly back at the wizard without the user having to aim.

---

[3] In our test, there was little to no emphasis on the collection of points, however, it was mentioned to the participants.

The Wisp

The wisp is implemented in UE5 as a simple object with very little logic controlling its actions. The appearance of the wisp was made using UE5's Niagara particle system, which allows us to create a glowing orb with trailing particles. When a wisp is spawned, it will locate the VR character in the VE and start floating toward it. Upon reaching a predefined distance from the character, the wisp will switch to an orbit state. In this state, it will circle the character at a predefined radius and speed. During this orbit, the wisp launches a projectile, at a random time, at the VR character. The wisp has no hitbox and can not be hit by the projectiles. Additionally, due to the primitive nature of the wisp's implementation, it may at any time pass through walls and sometimes even attempt to shoot at the character while inside walls.

The Wizard

The wizard is a distractor that we assembled using 3D models available on the UE5 marketplace and contains simple AI[4] behaviour trees[5] to control its actions. The default state of the wizard is "roam" In this state, the wizard will walk around in a specified area; for our tests, the area was set to the full extent of the map. The second state is "chase." This state is activated if the wizard has a line of sight to the user's avatar. In this state, the wizard will move towards the user as long as it maintains vision. When the wizard first gets a line of sight, it will play a sound to alert the user that it has been spotted. The next state is "attacking," which will happen once the wizard is close enough to the user's character. In this mode, the wizards will throw a projectile at the user every 2 seconds. After it has thrown, it will idle for those 2 seconds. If the user blocks the projectile thrown and the rebounded projectile hits the wizards, it will lose one health point and become stunned for 1.5 seconds. If two projectiles hit the wizard, then it will die. Upon death, a death animation will play, and the wisp will disappear after two seconds.

4.3.5 Additional extensions

The following minor sections will include information on the additional components in our implementation. They may not be as directly relevant to the task of Redirected Walking (RDW), but they are still components that are necessary for our implementation to be useful.

Save States and Save Data Manager

During the development of our system, we ran into an issue where our instance of UE5 would unexpectedly terminate due to a GPU crash. The

---

[4] AI in this context explains the behavior logic of video game non-player character and does not involve machine learning.

[5] See the UE5 documentation for further info add

issue became so severe that we implemented a save state into our system. With this save state functionality, we could quickly continue our tests in the event of a crash by having the state of the test saved. The save state was implemented using the standard UE5 *Save Game* framework and was controlled by our custom *Save Data Manager*.

When starting a new test, meaning the user's ID has changed, the Save Data Manager will generate a save for the new user. The save is then populated with the various parameters to be tested for each redirection technique. When a user enters a new test scenario, the Save Data manager will pick a random parameter from the save state and apply this to the scenario. To ensure that the user completes a test for a given parameter, the parameter is not removed from the save state until the test scenario completes. Thus, in the event of a crash, the parameter will still be present in the list of parameters and is re-tested after the crash.

Logging

Data collection is a vital aspect of user studies as it allows us to extract valuable insights about the behavior of our system and the users' interaction in the system. We implemented a comprehensive logging system that allows us to track and analyze events during users' tests and detect any odd behavior shown by our system. Moreover, by having a verbose logging routine, we may avoid re-running tests if we want to explore additional topics, as the required data may already be in the logs.

When development started, we intended to use the standard UE5 logging framework but found that it would be simpler, or of equal difficulty, to create a custom solution. The main draw to creating a custom solution was that we wanted to organize logs into folders based on user id and test scenario, which we found challenging with the standard approach. Our custom solution allows us to create logs pertaining to a unique user id, a unique VE, and a specific component of our RDW system. The logs are then organized in a file structure that reflects this order.

Log Type	Description
HMD Info	HMD-related variables such as position, rotation, and speed
Character Info	VR character-related variables such as position, rotation, and speed
Redirection Configuration	Configuration parameters for the redirection (e.g., current technique, current gain, etc)
Redirection Values	Values of the added rotation from the techniques that frame

Table 4.1: Information logged each frame by the RDW system

In our RDW system, the logging is handled in the Redirector Component, as it is connected to all other components in our system and thus has access to all the information in our system. When the redirector component is called every frame, it will first do what is required for redirection before logging relevant data. For every tick, we log the following information described in table 4.1.

Relative Actor Placement

To ensure the best starting position, we apply offsets to the button in the intermission room and add an offset to the rotation of the actor inside the level. In the intermission room, this is done simply by placing the button that brings the user to the next level, towards one of the corners of the RE. When the user presses the button, we calculate which direction the VR character should be looking upon entering the VE so that when the user turns to face the objective, they will be facing towards the center of the room[6].

Selection of Test Order

When running experiments, we need the tests to appear in a randomized order to remove any bias that the test order might have. To this end, we implemented a randomized system for choosing the test order. Whenever a test has been completed, that combination of parameter and VE is removed from the pool of remaining tests. The order is chosen on the fly as users load into the intermission level, where the scenario and parameter for the next test are chosen randomly[7] from the remaining pool of combinations. Once all tests have been completed, the system loads a final *credits* VE.

---

[6] For curvature gains, we have the user look slightly offset from the center due to the dead zone implemented in the technique

[7] This method uses UE5's integrated random number generator, which means that this is only pseudo-random, but we consider this to be acceptable for our use case.

Chapter 5

Experiment Setup

This chapter explains the specific details of the two user studies we conducted for this thesis. Following the methodology described in 3, we defined for each user study a set of parameters to be tested and a set of virtual environments they would be tested in.

5.1 User Study 1

The first user study focused on three redirection techniques: rotation, curvature and translational gains. For each of these we defined five parameters that we would tests, and built three purpose built VEs. The choice to only test five parameters were based on us wanting to limit the length of the study to aproxxamtly 25 minutes per person.

5.1.1 Choice of Parameters

The following paragraphs will explain the gains used during the first user study. A summary of the gains can be found in table 5.1.

For rotational gains, we saw that prior research showed a threshold of 1.24 to 1.27; our gains are based on this. We started our exploration from 1.25 and chose parameters at either side with 0.15 increments/decrements. As we were not exploring *lower gains*, and if we decremented twice, we ended up at 0.95, this gain was converted to the baseline parameter of 1.0.

Curvature gains were initially based on the same logic as rotational gains; however, during a pilot study, we found that much lower gains could be possible. Therefore, we used the lowest reported threshold, $11.5m$, in prior research as our starting point. We had previously tested with radii as low as $5m$ and wanted this as our most intense parameter. Therefore, we normalized the list of parameters to where we have $12.5m$ as our highest gain, and with $2.5m$ decrements, we end at $5.0m$. Additionally, the baseline parameter of ∞ is included.

For translational gains, we chose our parameters based on prior research and internal testing. Our initial pilot study did not include this technique, so we could not draw insight from this. Initially, we used a set the user feels threshold of 1.25, but how previously reported thresholds that this was too high. We, therefore, settled on using the reported threshold as an upper bound and decrementing by 0.15 until we reach our baseline parameter of 1.0.

RDW Technique	**Parameters**
Rotational Gain	1.0, 1.1, 1.25, 1.4, 1.55
Curvature Gain	∞, 12.5m, 10.0m, 7.5m, 5.0m
Translational Gain	1.0, 1.1, 1.15, 1.2, 1.25

Table 5.1: Parameters used in the first user study

5.1.2 Virtual Environments

In the first study, we created virtual environments for rotation, curvature, and translational gains. Per the methodology described for "Evaluating Redirected Walking" (Section 3.2.2), the goal of these VEs was to create general experiences that encouraged natural walking while maximizing the use potential of the redirection technique.

VE for Rotational Gains. The VE created for rotational gains, dubbed *Fire Drill*, was inspired by a similar VE from Razzaque's original work on RDW [34]. Our goal for this VE was to enable as much natural HMD rotation as possible, which is a key factor in applying rotational gains. To achieve this, we created a VE that presents the user with a hallway containing a set of buttons that the user must press to complete the test (Figure 5.1). The buttons are aligned in a zig-zag pattern, with buttons on either side of the hallway. The intended effect is that the user will press a button and then perform a turn/rotation of approximately 45 degrees in the VE to align themselves with the next button. We argue that the turn to the next button is a natural incentive to have the user rotate their head as the action aligns with the user's objective (of pressing all the buttons).

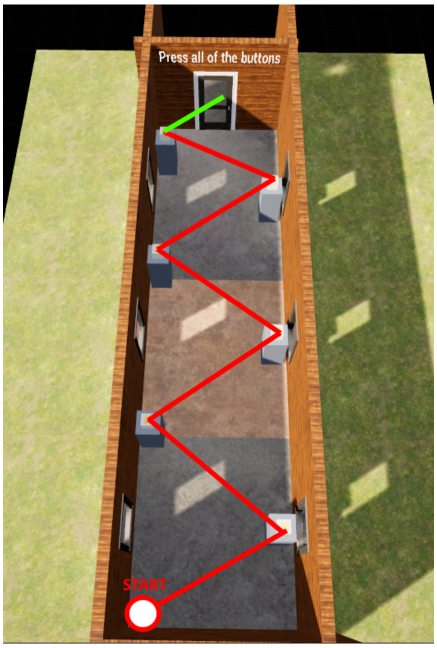


Figure 5.1: VE for rotational gain

VE for Curvature Gains. We designed the *Track and Field* VE for curvature gains testing. The VE draws inspiration from previous experiments that had the user walk in a straight line before reporting if they had felt any redirection. Our VE will transport the user into a stadium-like environment, with a running track and goal line in front of them (Figure 5.2). The goal for this VE was to have the user walk forward so that curvature gains could be applied. Therefore, we created a VE that has the user walk forward for a set distance before crossing the finish line and completing the test.

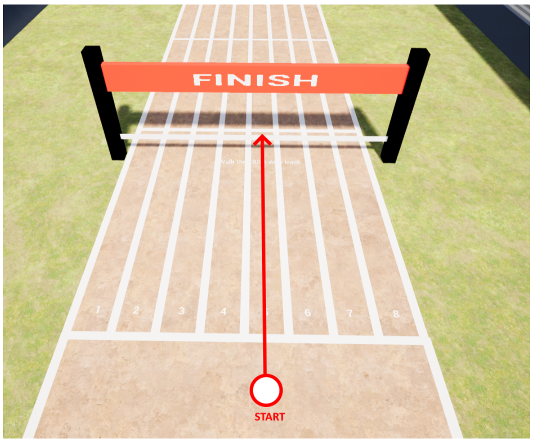


Figure 5.2: VE for curvature gain

VE for Translation Gains. We created the *Witch's Cauldron* to test translation gains (Figure 5.3). For this VE, our goal was to have the user walk in a virtual room slightly larger than the RE, which means that a higher translation gain will allow the user to explore more of the VE without being reset. We also wanted to include elements of fine-grained movement, as this could be an issue when movement scaling is high. Therefore, we created a VE that has the user complete a set of tasks within a room that is larger than the RE. The tasks are a set of fetch tasks that have the user collect colored spheres from various locations in the room before placing them in the center cauldron. The order of the spheres is randomized, and the next sphere in the order is displayed to the user above the cauldron. The test is over once the user has collected four spheres and placed them in the cauldron.

Figure 5.3: Translation gain VE

5.2 User Study 2

Our second user study focused on distractors' effects on rotational gains. To this end, we defined a set of parameters for this technique, created a custom VE for distractor testing, and created two unique diegetic distractors.

5.2.1 Choice of Parameters

The choice of parameters for this user study was similar to the choices for rotational gains in the first user study. However, different from the first study, we chose to select only four parameters, not five. This decision was made based on time constraints as we noted that the VE created for this test could take considerably longer than previously used VEs. We chose to keep the parameter 1.25 as a starting point and branched out as previously. However, from observations during the first study, we noted that the gains 1.4 and 1.55 received similar responses from participants, and thus we chose the middle ground of 1.5 instead. On the lower side, we kept the parameters 1.1 and 1.0.

RDW Technique	**Parameters**
Rotational Gain - Distractor	1.0, 1.1, 1.25, 1.5

Table 5.2: Parameters used in the second user study

5.2.2 Virtual Environments

To test the effect of distractors in the VE, we have created *The Dungeon.* This VE was designed with a higher level of immersion in mind and was made to accommodate diegetic distractors. The redirection technique that was tested in this VE was rotational gains, and thus we also included tasks that would force the user to look around and explore the VE. The VE's layout is a set of hallways arranged in a square with a cross-road intersection allowing traversal through the middle (Figure 5.4a). Scattered throughout the VE are four chests (Figure 5.4b), which serve as the task in the VE. The user was instructed to open all four chests and collect the gold found inside to complete the test. The location of the chests was randomized for each test to prevent the user from choosing an *optimal path* that could reduce the level of exploration needed[1].

(a) Dungeon VE layout (b) Dungeon VE atmosphere & chest

Figure 5.4: VE used for IRD evaluation

Additionally, the VE can be in one of two modes: with or without distractors. The effect of the distractors was tested by having the user complete two nearly identical tests, with the only difference being whether distractors were present. This meant that users would complete two tests using the same gain. To ensure that tests were as identical as possible, we included logic to place the chests within the VE in the same locations if the same gain was tested. The distractors we used for the tests are described in the following section.

5.2.3 Distractors

This section details the design choices for the two distractors we created for testing within The Dungeon VE. For an overview of the implementation details, refer to section 4.3.4.

- *Shared Aspects:* Although the two distractors differ in behavior and design, they share the ability to fire projectiles at the user. The

---

[1] Inspired by the naive search method employed by Peck, Fuchs and Whitton [33].

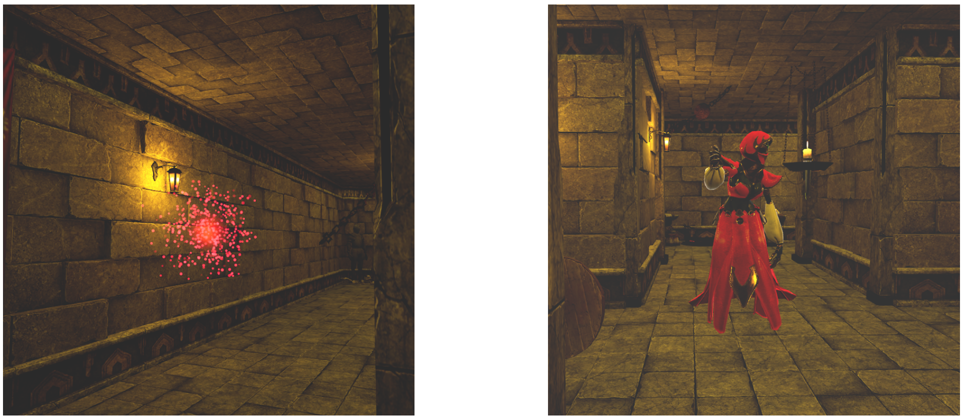

(a) The Wisp (b) The Wizard

Figure 5.5: Distractors used for IRD

projectiles fired will follow an arced trajectory toward the user. If the projectile impacts the user, then they will lose points. The user can block the incoming projectiles by using a shield affixed to their right controller. Any blocked projectile will bounce away from the user.

- *The Wizard* (Figure 5.5b) was included in this scenario to act as a way to increase the sense of presence that a user might feel in the scenario. The goal was that if users had a greater sense of presence, they would be less likely to notice that redirection was occurring. The wizard was designed as an explicit distractor, emphasizing it being strongly diegetic. In the VE, one wizard will always be actively roaming. If the user enters the wizard's field of view, it will approach the user before throwing a projectile at the user. If a projectile from the wizard is blocked, it will be bounced back toward the wizard, and the wizard will take damage. If hit by two projectiles, the wizard will perish, and a new wizard will spawn elsewhere in the VE. The intended response from the user to this distractor is that they should be wary of where it might be in the VE, and when it engages the user, it will induce rotation by having the user try to block the projectiles it throws.

- *The Wisp* (Figure 5.5a) is loosely based on the dragon which Peck, Fuchs and Whitton [33] used in their tests. The wisp is an explicit, diegetic distractor that is meant to force the user to do a large rotation in the VE. When spawned, the wisp will first approach the user, and upon reaching a refined orbit-radius, the wisp will travel on an orbital trajectory around the user. During this orbit, the wisp may, at any time, fire a projectile at the user. The wisp is only spawned if the user leaves a predefined *safe-area*, defined as a 1.5 radius from the center. To alert the user that the wisp has been spawned, it plays a spatially accurate audible sound. The intended response from the user is that upon hearing the noise, they will locate it in the VE and follow it with their gaze. As the distractor circles the user, this should induce a large amount of rotation.

Chapter 6

Results

The following chapter is divided into two sections, each detailing the results from one of the two user studies we conducted for this thesis. Following this chapter (Chapter 7) we discuss the various results and our thoughts around them. The design of the studies is described in section 3.2.2, and the procedure used is in section 3.3; for specific details of their setup, see section 5. Finally, for an insight into the methodology for data analysis, see section 3.2.4.

6.1 User Study 1

Our first user study focused on answering RQ_1 and RQ_2. Therefore, we explored the detection and acceptability thresholds for three redirection techniques: rotation, curvature, and translational gains. In addition, we also explored how specific types of redirection, or gains, affected the level of sickness that participants felt. Furthermore, we explored how different groups responded to redirection to uncover biases in demographic groups within our participants. Finally, we investigated the benefits of redirected walking in the VR experience.

After introducing the participants for this study, we will, in this section, show the analysis of the data we collected and present the results that the analysis yielded.

Participants. In this test, we had a total of 15 participants that completed the tests[1]. The participants were selected based on the procedure detailed in Section 3.3.1. The mean age among the participants was 25.6 (SD: 8.3), 60% being male and 40% female, with a mean height of 175.8 cm (SD: 10.3). Of the participants, none reported that they used medication for nausea, 26% used some form of vision correction and the mean self-reported balance ability was 4.0 (SD: 0.36). The mean self-reported VR experience of the participants was 2.0 (SD: 1.48), and their video game experience was 3.3 (SD: 1.65).

---

[1] Due to a technical issue, one participant could not complete all *curvature* tests and is therefore not included in those results

6.1.1 Reported Sickness

From our results, we are not able to say that there is any significant statistical difference between the reported sickness for the gains tested in this user study. We can however see that ANOVA reports there to be a statistically significant difference when grouping the reported sickness by the technique (Table 6.1).

Technique	df_1	df_2	F-value	F-tab	p-value
Rotational	4	70	1.609	2.50	0.181
Curvature	4	65	1.405	2.51	0.242
Translation	4	70	0.750	2.50	0.561
Between all	2	207	4.02	3.03	0.019

Table 6.1: ANOVA for reported sickness results within rotation, curvature and translation gain tests. Also compared between the three techniques. (Green p-values signify that we can reject H_0.)

The three techniques show a slight difference in mean reported sickness (Figure 6.1). Curvature gains (Mean: 0.04, SD 0.20) and translation gains (Mean: 0.01, SD: 0.11) are mostly equal, with curvature slightly above. However, rotational gains (Mean: 0.14, SD: 0.42) come ahead with a nearly quadrupled increase in mean reported sickness compared to the two other techniques.

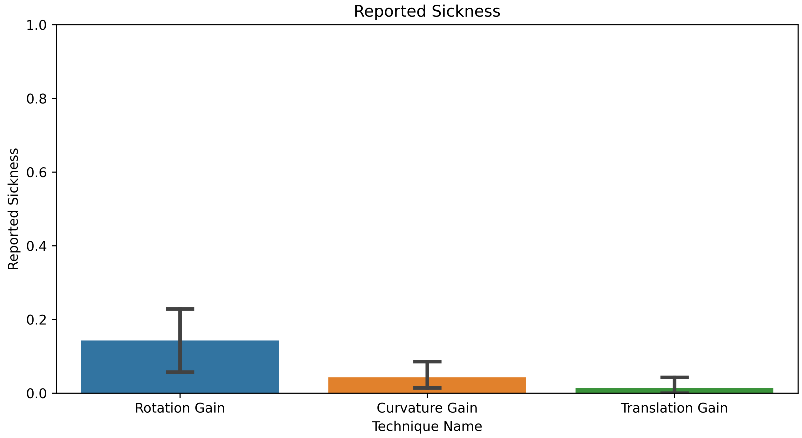


Figure 6.1: Reported Sickness means for the three techniques. (The y-axis of this figure has been shrunk for presentation. The original span of the y-axis is 0-4)

6.1.2 Detection Rate

For the three techniques we found that within the curvature gain and rotational gain tests, we were able to find a statistical significance between the parameters tested. We were however not able to determine this for translational gains. Results for this category are further discussed in section 7.2.1.

Technique	df_1	df_2	F-value	F-tab	p-value
Rotational	4	60	3.3895	2.52	0.0136
Curvature	4	56	3.0396	2.08	0.0233
Translation	4	60	0.1707	2.52	0.9526

Table 6.2: ANOVA for detection of redirection for user study 1. footnotesize(Green p-values signify that we can reject H_0.)

Rotational Gains. Detection rates for the five gains tested show a rising trend as the gains rise in intensity (Figure 6.2). However, with a 10% detection threshold, only the baseline parameter can be considered subtle.

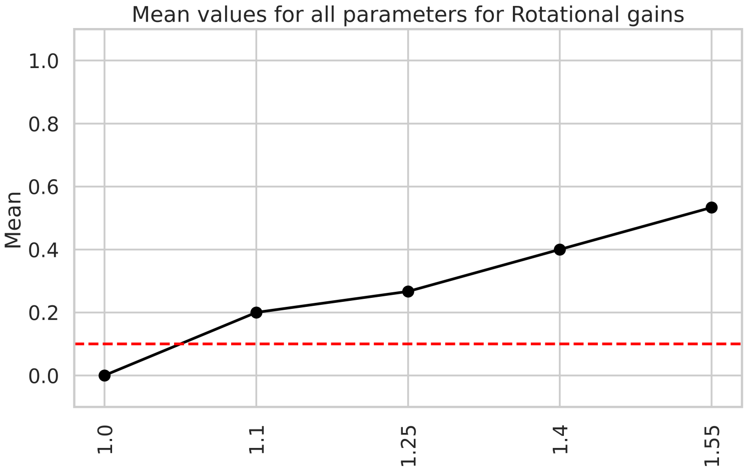


Figure 6.2: Mean reported detection rate of rotational gains for each parameter.
(Red bar signifies 10% threshold)

Curvature Gains. The mean detection rate for curvature gains (Figure 6.3) shows a similar pattern to rotational gains, where we see an increasing amount of detection as we move toward a more intense gain. In contrast, however, two parameters (12.5 and 10.0) fall under the defined detection threshold. Surprisingly, the baseline parameter, ∞m, appears above the 10% threshold as two participants reported that they detected redirection during this gain.

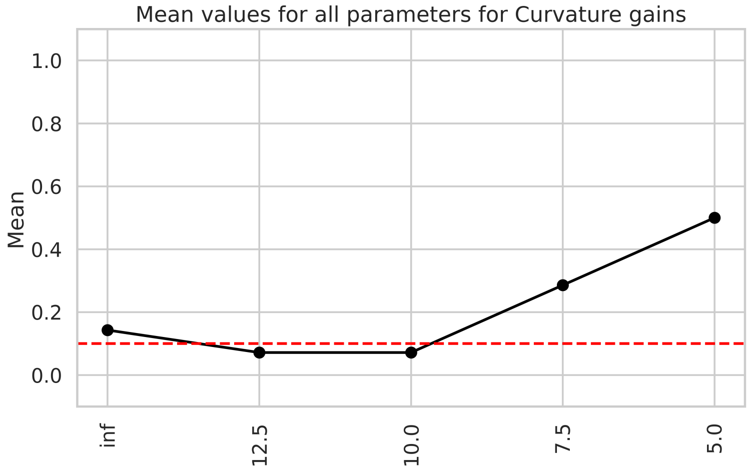


Figure 6.3: Mean reported detection rate of curvature gains for each parameter.
(Red bar signifies 10% threshold)

Translational Gains. Translational gains showed no statistical significance between the parameters. From the reported means, we see that, except for the highest gain, all parameters stay below the detection threshold of 10% (Figure 6.4). However, as we do not find statistical significance, these results can only be used for speculation.

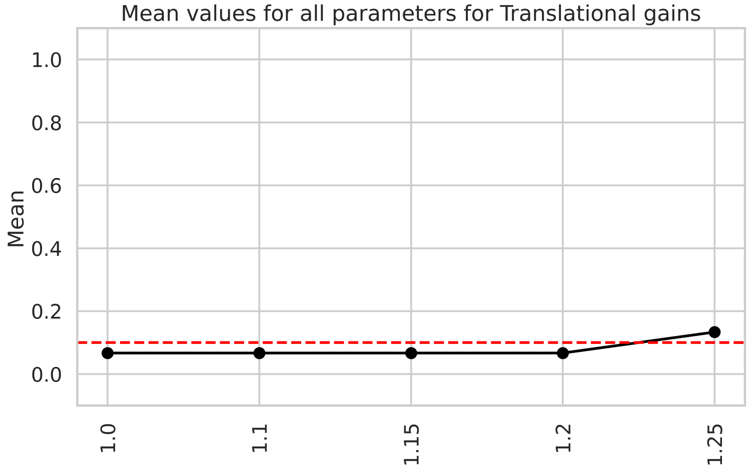


Figure 6.4: Mean reported detection rate of translational gains for each parameter.
(red bar signifies 10% threshold)

6.1.3 Acceptability

As with detection rates, we can show a statistically significant difference in the mean acceptability ratings within rotational gains and curvature gains.

Similarly, we could not determine whether there was a difference in the means for translational gains. Acceptability results are discussed in section 7.2.2.

Technique	df_1	df_2	F-value	F-tab	p-value
Rotational	4	60	2.5370	2.52	0.0475
Curvature	4	56	2.9277	2.53	0.0274
Translation	4	60	1.0600	2.52	0.3829

Table 6.3: ANOVA for acceptability of redirection for user study 1.
(Green p-values signify that we can reject H_0.)

Rotation Gains.

The mean reported acceptability (Figure 6.5) shows that for the five gains; there is a trend of worsening acceptability as the gain intensity rises. With the 4.0 threshold, only the most intense parameter counts as unacceptable.

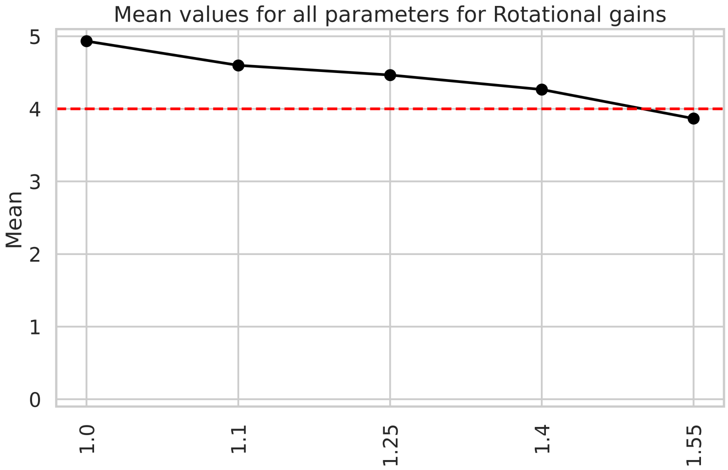


Figure 6.5: Mean reported acceptability of rotational gains for each parameter
(Red bar signifies 4.0 threshold)

Curvature Gains. Mean reported acceptability for curvature gains (Figure 6.6) shows that once the gain passes 10.0m, acceptability falls. However, with a 4.0 threshold, no gains result are classified as unacceptable.

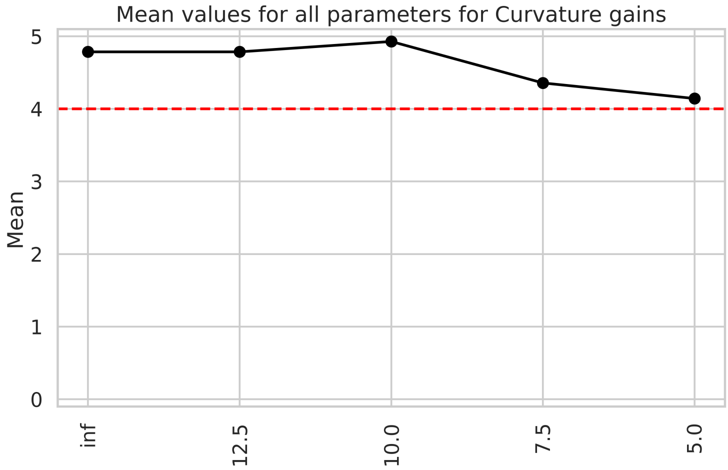


Figure 6.6: Mean reported acceptability of curvature gains for each parameter.
(Red bar signifies 4.0 threshold)

Translational Gains. Translational gains showed no statistical significance between the parameters. However, from the reported means (Figure 6.7), we see that all parameters were within the threshold for being considered acceptable.

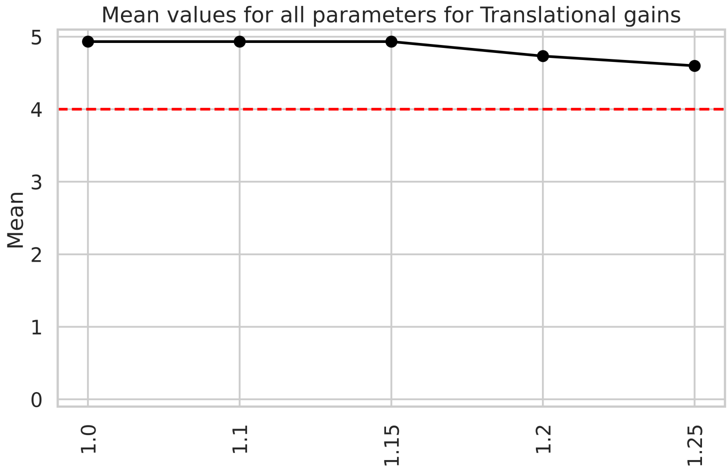


Figure 6.7: Mean reported acceptability of translational gains for each parameter.
(Red bar signifies 4.0 threshold)

6.1.4 Effect of Parameters

To measure the effectiveness of parameters we investigate the quantitative metrics gathered during tests. The metrics that we investigate are the number of resets, distance traveled between resets, rotation added, and distance from the center. The results presented here are discussed in section 7.2.3.

In the following illustrations, the actual parameter values have been replaced with generic names to allow plotting within the same figure. Table 6.4 shows the relation to actual values.

Technique	P1	P2	P3	P4	P5
Rotational	1.0	1.1	1.25	1.4	1.55
Curvature	inf	12.5	10.0	7.5	5.0
Translation	1.0	1.1	1.15	1.2	1.25

Table 6.4: Translation table for parameter values in result graphs

Added Rotation

To validate that the various parameters for the three algorithms actually show a difference in behavior we inspect the *average rotation added per second*[2]. In both algorithms, we can see a clear trend that shows the different intensities change the amount of rotation applied (Figure 6.8).

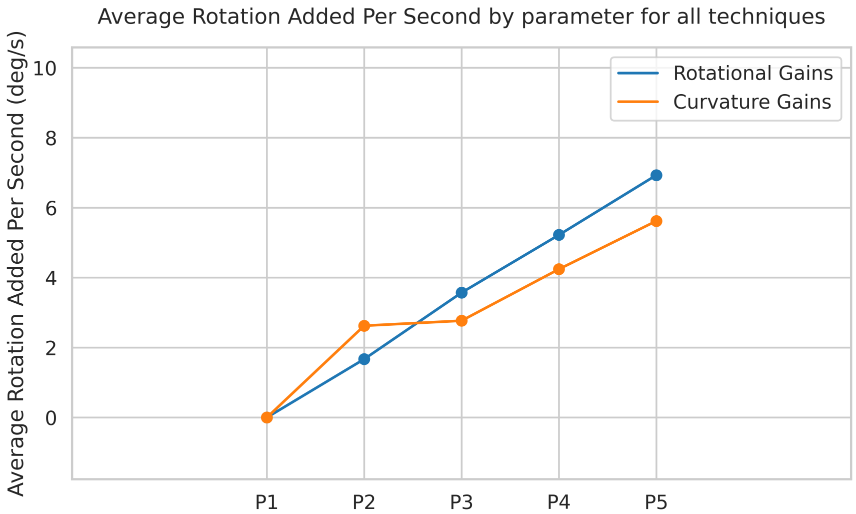


Figure 6.8: Average rotation added per second for rotational and curvature gains

Number of resets

In our studies, we were not able to show that differences in parameter intensity had any considerabøe affect on the number of resets when using rotational or curvature gains. The results did however show a clear trend of users requiring fewer resets as the translation gain rises (Figure 6.9).

---

[2] We do not consider translation gains here as it does not use rotation as a metric.

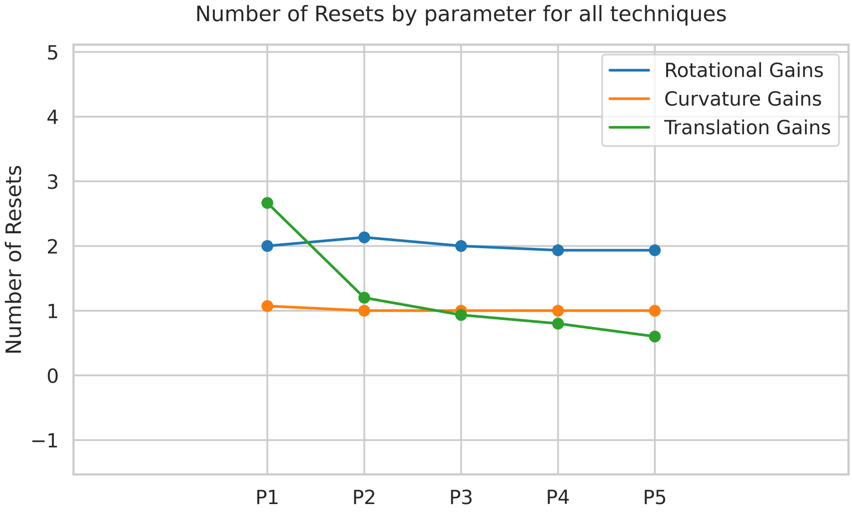
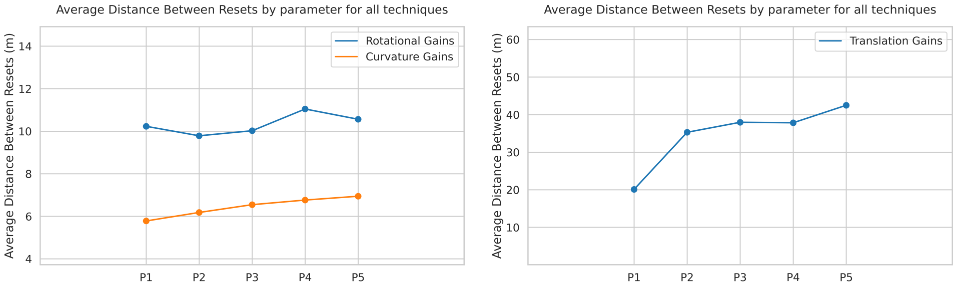


Figure 6.9: Number of resets for rotational, curvature, and translation gains

Distance Traveled Between Resets

In both curvature (Figure 6.10a) and translational (Figure 6.10b) gains, we can see increases in the distance a participant could travel between resets. Rotational gains partly show this trend, but it appears less evident. Since the translation gains tests had the user walk much further than in rotation and curvature tests, we have split them into separate tables.



(a) Average distance between resets for rotational and curvature gains

(b) Average distance between resets for translation gains

Figure 6.10: Distance between resets for rotational, curvature, and translation gains

Distance To Center

With the exception of a small jump between P1 and P2, we can see from our results that for rotational gains (Figure 6.11) there is a trend of participants spending more time closer to the center as the parameter value rises. This is not evident for curvature gains where we actually see a reverse trend where users seem to be farther away from the center as the parameter increases in intensity. Translation gain was not considered as we used a static gain, and the center was not relevant when applying the gain.

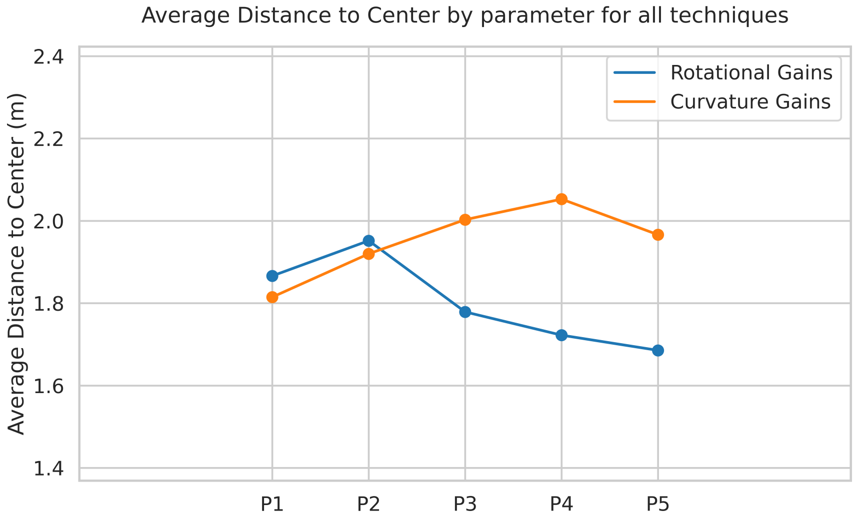


Figure 6.11: Average distance to center for rotational, curvature, and translation gains

6.1.5 Participant Attributes

For all attributes that the participant supplied, we ran a Kruskal-Wallis test to determine if there was any significance in the groups that existed within the participants. The groups were first explored for detection rates (D) and acceptability (A) across all tests (Table 6.5). In addition to exploring if groups had a general effect on the detection rates and acceptability, we also explored if there were any differences when only considering *Baseline tests without redirection* (B), and *Tests with active redirection* (R) separately (Table 6.6). Groups that contained fewer than two participants, and attributes with two or fewer groups, were excluded from the check.

Parameter	Attribute	n-groups	K-Stat	p-value
A	Gender	2	2.06	0.1511
A	Age Group	2	2.98	0.0844
A	Height Group	3	7.69	0.0214
A	VR Proficiency	2	14.92	< 0.01
A	Video Game Proficiency	2	6.03	0.0141
A	Balance Ability	< 2	NaN	NaN
D	Gender	2	0.38	0.5386
D	Age Group	2	0.01	0.9248
D	Height Group	3	1.84	0.3986
D	VR Proficiency	2	16.71	< 0.01
D	Video Game Proficiency	2	4.01	0.0452
D	Balance Ability	< 2	NaN	NaN

Table 6.5: Kruskal-Wallis test for participant attributes.
(Orange group number signifies that groups were removed due to low amount of participants. Green p-values signify that we can reject H_0.)

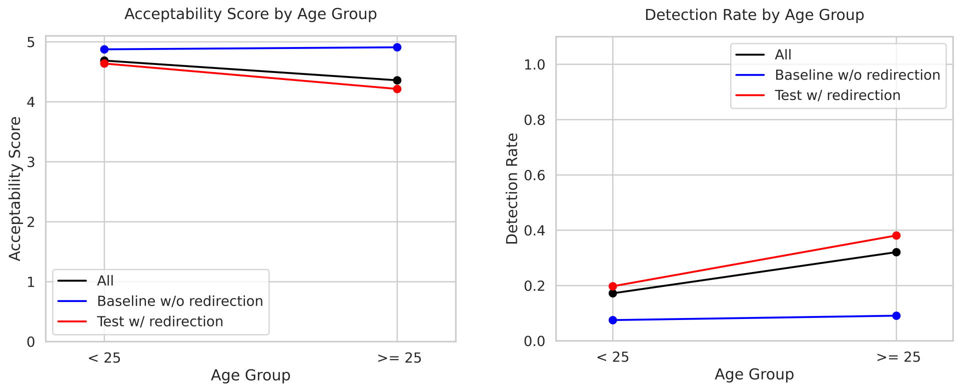
The attributes that showed to have a p-value of less than 0.05 (meaning 95% confidence), we further delve into the results of the groupings. Groups that show no significance, or those that could not be compared due to lack of participants, are not considered.

Type	Parameter	Attribute	n-groups	K-stat	*p*-value
B	A	Gender	2	2.24	0.1347
B	A	Age Group	2	1.16	0.2819
B	A	Height Group	3	3.99	0.1359
B	A	VR Proficiency	2	0.41	0.5227
B	A	Video Game Proficiency	2	0.72	0.3947
B	A	Balance Ability	< 2	NaN	NaN
B	D	Gender	2	0.00	0.9757
B	D	Age Group	2	0.85	0.3578
B	D	Height Group	3	0.02	0.9916
B	D	VR Proficiency	2	0.73	0.3915
B	D	Video Game Proficiency	2	0.25	0.6167
B	D	Balance Ability	< 2	NaN	NaN
R	A	Gender	2	1.14	0.2857
R	A	Age Group	2	4.63	0.0314
R	A	Height Group	3	7.79	0.0203
R	A	VR Proficiency	2	15.60	< 0.01
R	A	Video Game Proficiency	2	8.26	< 0.01
R	A	Balance Ability	< 2	NaN	NaN
R	D	Gender	2	0.41	0.5210
R	D	Age Group	2	0.16	0.6886
R	D	Height Group	3	1.99	0.3705
R	D	VR Proficiency	2	16.49	< 0.01
R	D	Video Game Proficiency	2	5.25	0.0220
R	D	Balance Ability	< 2	NaN	NaN

Table 6.6: Kruskal-Wallis test for participant attributes with filtering on base-level tests.

(Orange group number signify that groups were removed due to low amount of participants. Green p-values signify that we can reject H_0.)

Age Group. For examining age groups we defined two groups *25 and above* and *Below 25*. Within the age groups, we found that there is a significant difference in the non-base-level tests for both acceptability and detection rate. We saw that the younger age group in general rates the experiences to be more acceptable (Figure 6.12a), and detects fewer gains (Figure 6.12b), than the age group above them.

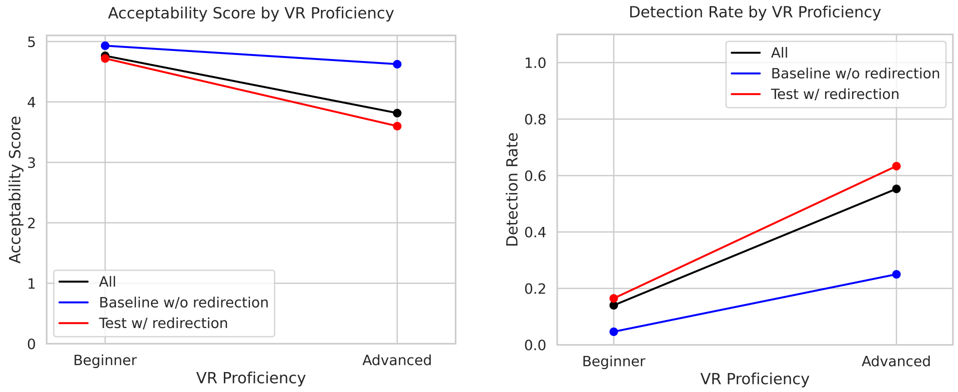


(a) Acceptability by age group

(b) Detection rate by age group

Figure 6.12: Age group means for acceptability and detection rate

VR Proficiency. For VR proficiency we found statistical significance for both detection rate and acceptability when comparing all tests, but when separating into baseline and redirected tests we do not find any significance in acceptability. We saw that those who considered themselves more adept with VR detected more gains (Figure 6.13b), and had lower acceptability when exposed to redirection (Figure 6.13a).

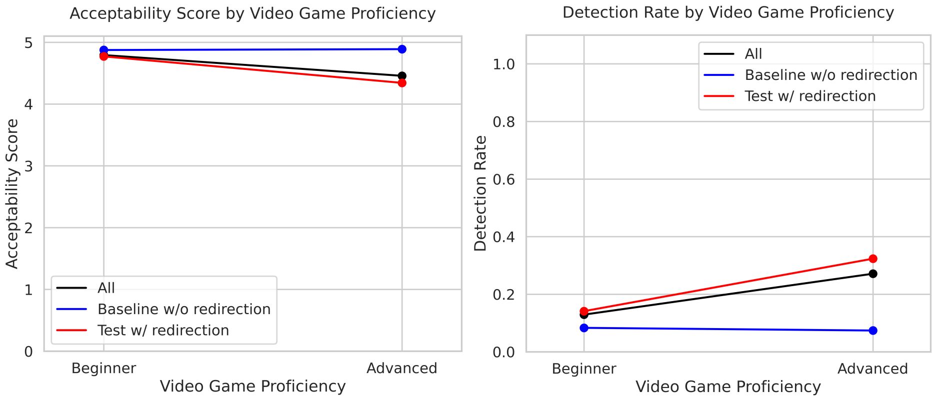


(a) Acceptability by VR Proficiency

(b) Detection rate by VR Proficiency

Figure 6.13: VR Proficiency means for acceptability and detection rate

Video Game Proficiency. We were not able to show any significant statistical difference in detection rate for video game proficiency, but we did find it for acceptability. In figure 6.14a we can see that there is a slight shift in acceptability where users with a higher proficiency rate experiences with redirection less acceptable.

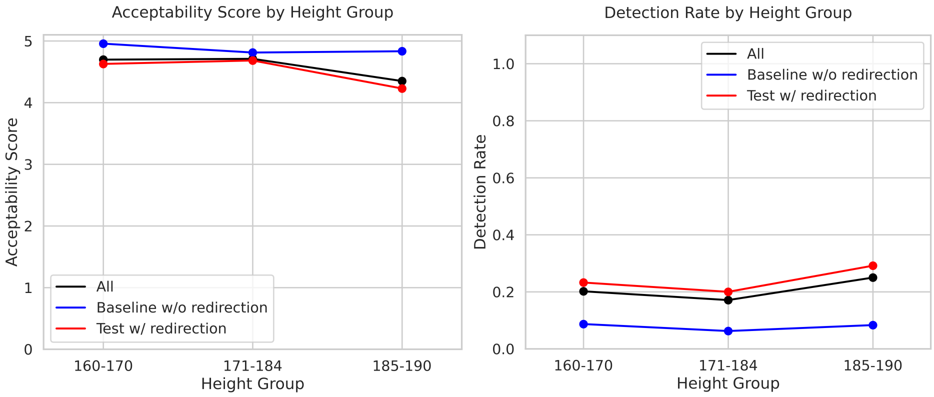


(a) Acceptability by Video Game Proficiency

(b) Detection rate by Video Game Proficiency

Figure 6.14: Video Game Proficiency means for acceptability and detection rate

Height. Within the height groups, we found there to be statistical significance in acceptability but not detection rate. The results showed that participants 185-190 cm group rated the tests with redirection lower than those who were lower (Figure 6.15a).



(a) Acceptability by height group

(b) Detection rate by height group

Figure 6.15: Height group means for acceptability and detection rate

6.2 User Study 2

In our second user study, we placed a stronger focus on distractors, and how the presence of distractors could affect redirected walking. More specifically we sought to investigate how the detection thresholds for rotational gains might be shifted by having distractors present in the VE. Additionally, we wanted to explore what effect distractors would have on the effectiveness of rotational gains. With this second user study, we aimed to answer RQ_3 and RQ_4.

As we only test a single technique in this study, we will not compare how techniques differed but rather focus on how the response of participants changed dependent on whether or not distractors were present in the VE.

Participants. In this test, we had a total of 15 participants that completed the tests. The mean age among the participants was 24.9 (SD: 6.3), 86% being male, and 14% female, with a mean height of 177.8 cm (SD: 9.1). Of the participants, none reported that they used medication for nausea, 6% used some form of vision correction and the mean self-reported balance ability was 3.5 (SD: 1.7). The mean self-reported VR experience of the participants was 2.1 (SD: 1.3) and their video game experience was 3.3 (SD: 1.7).

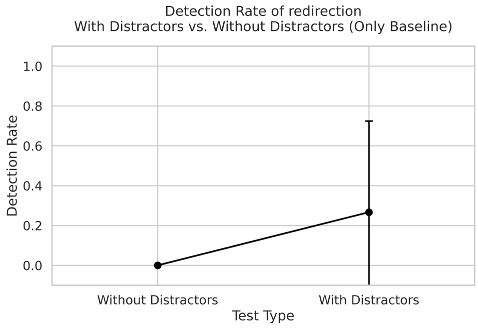
6.2.1 Detection Rate and Acceptability

To investigate if the presence of distractors had any significant effect on the detection rate or acceptability rating, we used ANOVA (Table 6.7). The two group means we investigated were tests that had distractors, and those that did not. Additionally, we looked at if there was any difference when we analyzed all test cases, only baseline tests without redirection, and tests that had active redirection. In most cases, we were not able to find any statistical significance in the data, except for detection rates among the baseline tests.

Tests	Type	df_1	df_2	F-value	F-tab	p-value
All	D	1	118	0.04	3.9214	0.8240
All	A	1	118	0.05	3.9214	0.8093
B	D	1	28	5.09	4.1959	0.0320
B	A	1	28	1.43	4.1959	0.2415
R	D	1	88	0.54	3.9493	0.4603
R	A	1	88	0.01	3.9493	0.8985

Table 6.7: ANOVA results for detection rates for rotation gain for distractor testing. The two groups being compared are tests with distractors and those without distractors.
(Green p-values signifies that we can reject H_0 with 95% confidence or more)

Base Tests. For the baseline tests, where no redirection occurred, we see that participants were more likely to report that they detected redirection if distractors were present (Tabel 6.16).



(a) Means for detection rate between tests with and without distractors

Group	Mean	Std.
Without Distractors	0.000	0.000
With Distractors	0.266	0.457

(b) detection rate group Means

Figure 6.16: Means for detection rate between tests with and without distractors

6.2.2 Effect of Distractors

To check the effectiveness of distractors, we investigate some of the same metrics as with general RDW. However, we emphasize the average HMD rotation and added rotation, as this is what we sought to increase with our distractors.

Duration

Although not applicable to the first user study, we included an insight into the duration for distractor testing. We can see from the results (Figure 6.17) that introducing distractors generally caused participants to spend longer completing their tasks.

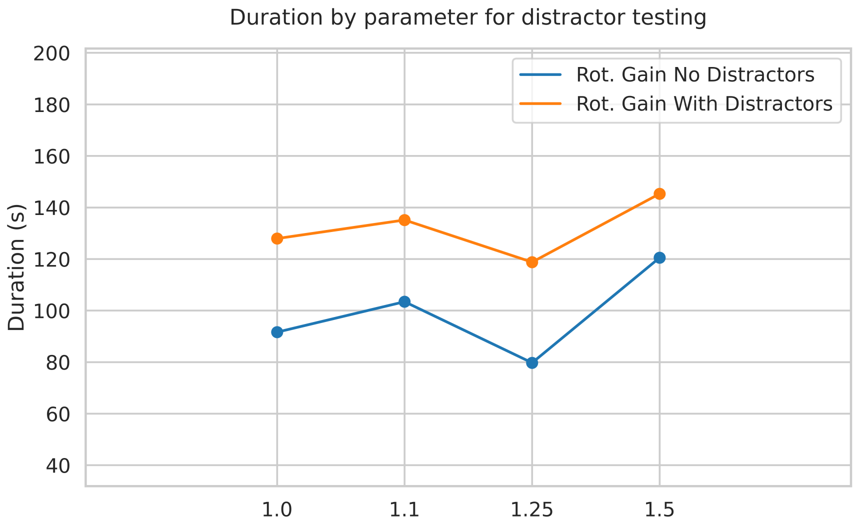


Figure 6.17: Duration of tests for distractor testing

Number of Resets

The number of resets showed that users reset more often when distractors are introduced (Figure 6.18). These results are, however, not based on time, and the increase in resets is likely due to an increased duration of the tests. Looking at the trend of the two scenarios, we can see that they are similar.

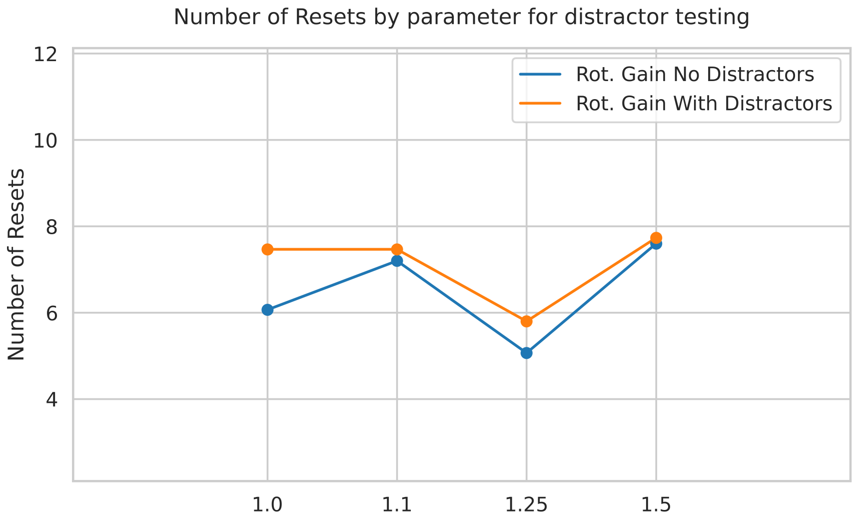


Figure 6.18: Number of resets for distractor testing

Distance Traveled Between Resets

The distance between resets *is* bound to time, and thus we can directly compare the means of the two scenarios. Figure 6.19 illustrates that the distance between resets remains relatively equal between the two scenarios, with a slight up-tick for distractors during some parameters.

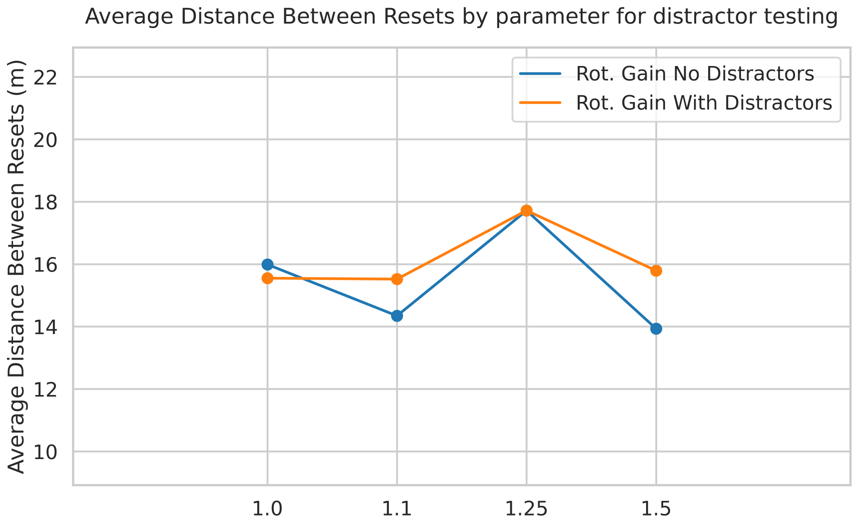
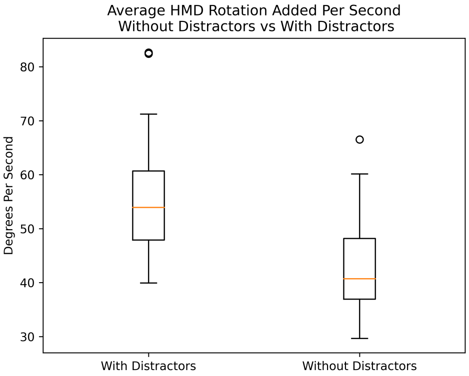


Figure 6.19: Average distance between resets for distractor testing

Average HMD Rotation Per Second

The average HMD rotation represents how much the users had to turn their heads during the tests. In figure 4.3.4, we show that the total amount of HMD rotation increased substantially during tests that contained distractors.



(a) A plot of average HMD rotation per second

Group	Mean	Std.
Without Distractors	42.771	8.771
With Distractors	54.754	9.800

(b) Means of average HMD rotation per second

Figure 6.20: Comparison of average HMD rotation during tests with distractors vs. tests without distractors

Average Added Rotation Per Second

Additionally, we look at the average rotation added per second by the rotational gain technique. Here we explore how much extra rotation is applied for each of the four parameters tested[3]. To determine if there was any difference between the added rotation when grouping by the parameter and if there were distractors in the VE, we use a Multivariate-ANOVA (Table 6.8). The analysis showed great confidence in comparing the added rotation when grouping by parameter and *distractor-state*. In table 6.9, we can see that introducing distractors in the VE allowed for an increase in average added rotation of 43% to 13%.

Type	df_1	df_2	F-value	F-tab	p-value
C(Distractor):C(parameter)	7	112	55.9	2.0923	< 0.01

Table 6.8: MANOVA for HMD rotation for tests with and without distractors

---

[3] Note that the *lower* gain is constant at 0.95

Parameter	No Distractor	With Distractor	% Difference
1.0	0.930	1.338	43.8
1.1	3.087	3.966	28.4
1.25	6.404	7.253	13.2
1.5	9.377	11.517	22.8

Table 6.9: Comparison of average rotation added per second for test with and without distactors

6.3 Summary

In this chapter, we have presented the findings from both of our user studies. In this section, we will summarize key findings for interpretation and discussion in the next chapter.

Detection Thresholds and Acceptability Thresholds

The following table (Tabel 6.10) summarizes the findings for detection and acceptability for rotational, curvature, and translational gains.

Metric	**Technique**	**Threshold**
Detection	Rotation	1.0 - 1.1
Detection	Curvature	10m - 7.5m
Detection	Translation	Inconclusive
Detection	Distractors	Inconclusive
Acceptability	Rotation	1.4 - 1.55
Acceptability	Curvature	5m -
Acceptability	Translation	Inconclusive
Acceptability	Distractors	Inconclusive

Table 6.10: Summary of acceptability and detection thresholds.
(Thresholds are represented as a span of two values. Where the left side is acceptable/subtle and the right is unacceptable/overt)

Redirection Effect

In the following table (Table 6.11), we summarize the effects the three redirection techniques had during the tests during the first user study. The *effect* column explains what happened as the parameter intensity grew.

Technique	**Metric**	**Effect**
Rotation	Number of resets	No Effect
Curvature	Number of resets	No Effect
Translation	Number of resets	Fewer resets
Rotation	Distance between resets	Little Effect
Curvature	Distance between resets	Slightly longer distance
Translation	Distance between resets	Longer distance
Rotation	Distance to center	Less distance
Curvature	Distance to center	More distance
Translation	Distance to center	Not applicable

Table 6.11: Summary of redirection effects for first user study

Further, in tabel 6.12, we summarize the effects of having distractors present in the VE. In this table, the effect column describes the effect of adding distractors to the VE.

Metric	**Effect**
Duration	Longer Duration
Number of resets	No change
Distance between resets	No Change
Avg. HMD rotation	Large increase
Avg. Added rotation	Increase

Table 6.12: Summary of redirection effects for second user study

Chapter 7

Discussion

The collected data from the two user studies provided us with a valuable data set that helped us evaluate Redirected Walking (RDW) and RDW with distractors. This chapter is divided into a section containing general remarks that are applicable to both user studies, a section for each of the user studies, and finally, a summary that discusses how our results answer the research questions set for this thesis.

7.1 General Remarks

Although we believe that the data collected during both of our user studies are valid, and can be used to answer our research questions, we have some general remarks that were observed during the testing process. The remarks may not be directly related to the final conclusions, but it is worth being aware of them as some may have had some unseen effect on the results.

Redirection Controller

We noted fairly quickly in our pilot study and later during our main studies that our redirection controller could not redirect users in a way that worked within the Real Environment (RE) we had available. Our results show that neither curvature nor rotational gains lower the number of resets the user experienced during a test scenario. Additionally, the distances between resets were also only minimally affected by redirection.

A reason why the controller was not able to redirect effectively could be that the RE that we used was not large enough to accommodate the algorithm used. It was observed during the tests that users were being redirected toward the center of the room, but the rate at which this happened was not enough to avoid hitting the wall and stopping a reset. This behavior was especially apparent for curvature gains where the user's path was clearly curved towards the center. However, as all of the curvature gains we tested had a wider radius than the half-length of the RE, it was impossible to complete a full orbit without hitting the boundary and triggering a reset.

A second design choice that was a possible flaw was how the controller was to some degree *limited* for rotational gains. In most applications of rotational gains, the redirection controller can choose a gain between a *lower* and *upper* gain. The lower gain states how much the controller can reduce angular velocity, while the upper gain states how strongly the controller can amplify the rotation. By switching between the two gains, the controller can reorient the user independently of where they face and rotate. In our implementation, however, we decided that testing both upper and lower and upper gains simultaneously would result in a huge number of tests if we were to test all permutations of gains. The total rotation gain tests would increase by five for every additional lower gain tested. If we were to test five lower gains, we would have had $5 * 5 = 25$ rotational gains tests, adding over 15 minutes to the testing time, which many participants would not have accepted. Therefore, we only tested the upper gains during our studies, which may have directly lowered the utility of rotational gains for redirection.

Although we considered that these flaws could be present in our controller during the development phase, the lack of a large RE for testing resulted in little full-scale testing. This meant that we could not fully confirm if our redirection controller could redirect in a way that benefited the user. However, even though the controller may not have been redirection optimally, this did not interfere with the research focus of this thesis, and can we still derive valid conclusions.

Inconsistent or Unclear Instructions

We observed that between users, there existed a significant difference in how users rated the various experiences during the study. Therefore, there may have been a lack of or inconsistent instructions given to the participants before and during the tests. Specifically, we noted that some users quickly rated a sub-par experience with a low acceptability score, while others never rated below the middle of the scale. Of course, we cannot judge how a user *should* feel about a test. However, we found it odd that two users could describe their experience very similarly but then rate them so differently regarding acceptability.

This possible fault in the procedure could have affected the data collected. However, as we are dealing with personal experiences, it is not possible for us, in retrospect, to determine if the users were misinformed about the scoring procedure or if they were experiencing the tests differently, which would warrant the different scores. Still, there should have been a more precise explanation of the scoring procedure to eliminate doubt about how users score the tests.

Testing Environment/Location

The RE in which we conducted our user studies had some major flaws we could not eliminate. As a result of testing in an active university building, the location was not well isolated from sound and disturbances.

Specifically, considering how the sense of presence, or immersion, is a key element in VR, some elements of the RE could have reduced the user's sense of presence, ultimately reducing their acceptability rating for that experience. Firstly, people sometimes walked past the testing location while talking loudly, which the user could hear. The noise was, for the majority of the time, at a negligible level that the speakers of the HMD were able to hide. There were, however, times when the volume and amount of background noise would be audible to the participant. Secondly, the floor of the RE was not entirely uniform, with a deformation in the corner and a carpet at the edge of one of the sides. Although all users did not encounter these deformations, some reported that the change in the floor surprised them and that they had to pause briefly to *re-immerse* themselves in the VR experience.

The noise events generally did not last for very long but could sometimes last a duration covering one or two test cases. Therefore, as the noise level was not constant for participants, this could have introduced biases for some tests where the noise was not non-negligible. Additionally, as we noted, only a handful of participants came into contact with the deformations of the floor, which may have introduced a second bias. However, as we did not measure the change in background noise or track events where the user encountered deformations during the tests, we can not say if this had a measurable effect.

Resets

During the tests, some users reported that resets were a key factor in drawing away immersion. In many instances, users stated that they became disoriented following a reset. This response became seemingly less present in the participants after having been in the Virtual Environment (VE) for some time. However, some participants still reported that the resets did affect the experience quite negatively.

It was expected that resets would happen during our testing. However, we did not foresee some users' responses, as during internal testing, the reset mechanism became second nature to us. Furthermore, in some instances, we had participants that were *unlucky* with their resets, which caused several to happen resets in quick succession. This could, for example, be the case if the user was reset in the corner of the RE but at a turning point in the VE, causing them to be reoriented by an initial reset, before then immediately turning around and walking into the boundary again. To mitigate this effect, one would have to consider the VE when attempting to reset, but as we employed a simple generalized controller, this was not done in our implementation.

Network Delay

Another contributing factor to how the participants perceived the experience could have been biased by the network delay introduced using a wireless Head-Mounted Display (HMD)[1]. Although we did not notice any visible latency or delay amongst ourselves, two participants noted this. One participant was part of the pilot test, and the other was in the first user study, but both were rated as very experienced with VR. Upon some questioning on their VR usage, both participants stated that they used a tethered HMD, which could allude that using the wireless setup introduced additional latency to the experience, which could have affected the participant's experiences. The effect this latency might have caused is unclear; however, given that all tests were done with the same setup, the results should still be valid. Furthermore, as only two participants noticed any latency, of which both stated that experience was still acceptable, we do not consider this to have had a strong effect on our research. However, to determine how latency may affect the VR experience, a follow-up study should explore how users experience introduced latency of varying degrees.

Level Of Discomfort/Cybersickness

Note: *The focus of our studies was not to explore cybersickness and the prevalence of this during redirection. However, it was monitored as a possible side effect. Participants in our studies were clearly informed that they should either pause or cancel the testing if they began to feel ill.*

Our results showed that within techniques, we could not determine any difference in how users were affected in terms of reported sickness. We were only able to show that there was a difference in how the three different techniques affected the users. We did, however, see a trend in the means of single techniques that implies a higher gain would lead to a greater level of sickness; this, however, can not be confirmed without further study. In contrast, there was statistical significance when comparing the three redirection techniques. We could show that rotational gain significantly increased reported sickness compared to curvature and translation gain.

We believe we did not see statistical differences within single techniques because of the low number of people who reported any sickness. This effect is likely because cybersickness has been shown to affect users differently; thus, not all our participants were affected. Moreover, the length of exposure to redirection may have had an effect, and due to the short test durations, the users may not have been exposed long enough to develop symptoms.

Between the three techniques, we can show that rotational gains caused a higher level of cybersickness than curvature and translation gains. We have no definite answer to this, but we can speculate on some potential reasons. First, as will be discussed later, translation gains may have been tested with too low gains. Thus an increase compared to translation gain could be expected, given that rotational gains had gains that users deemed unacceptable. Second, compared to curvature gains, the duration

---

[1] Note that the HMD used (Oculus Quest) is a slightly older HMD, and that newer HMDs may not have this problem.

of rotation tests was considerably longer (4̃0 seconds vs. 1̃5 seconds), which may have caused users to be under-exposed during the curvature tests. Third, it is possible that the gains tested during rotation tests were not scaled correctly compared to the other techniques and that the intensity was higher when subjectively compared. Finally, it is also possible that rotation gains are more exposed to causing cybersickness.

7.2 User Study 1

This section discusses the results gathered and analyzed for our first user study in an exploratory manner to uncover why they showed what they showed.

7.2.1 Detection Thresholds

Our results reveal, with statistical significance, that the detection of rotational gains increases as the parameters become more intense (Figure 6.2), which is consistent with our initial expectations. Nevertheless, there is a discrepancy between our findings and previous research regarding the detection threshold of rotational gains. Previous research has suggested a threshold close to 1.25, whereas our results indicate that even a gain of 1.1 exceeds the limit set for detectability.

It is plausible that the static method utilized in our study to determine the detectability of a technique was suboptimal and may have misrepresented the actual threshold. We observed that participants often paused to recall whether they had felt any unnatural sensations during the previous test scenario, as detectability was assessed after the completion of each test scenario.

Additionally, we observed that the parameters considered overt varied significantly among participants. The findings support this observation, suggesting a correlation between particular participant attributes and an increased detection rate of redirection (Section 6.1.5). Our participant sample predominantly consisted of students from the informatics institute, with limited representation from other demographic groups. Consequently, shared biases among participants might have influenced the outcomes. The effect of participant groups is further discussed in section 7.2.4

In the case of curvature gains, our results suggest the existence of a detection threshold between a radius of 10 meters and 7.5 meters. As with rotational gains, we observed the anticipated trend of more intense variables being detected more frequently. Interestingly, over 10% of participants rated the baseline test without redirection as detected. To this, one participant claimed to feel a sensation of being *slowed down*, as if a negative translation gain was applied, which was not the case. This sensation might have resulted from the participant's exposure to a translational gain test with higher parameters immediately before the curvature gain test, potentially altering their perception of speed and making them feel as if they were walking slower than expected.

Nevertheless, without further research, it is difficult to say whether this observation is a coincidence or indicative of an actual correlation between test order and test results.

Lastly, for translational gains, we were not able to find any statistical significance within the results. We believe this to be a result of improper parameter choices. In that, we chose too weak parameters and thus did not confidently pass the detection threshold. However, by looking at the results, we can see a slight uptick in detection once the parameters reach 1.25, which is close to the thresholds reported by previous research (Table 2.1). Had we followed similar guidelines as the other two techniques and not based it on personal experiences, we may have chosen parameters that presented a more precise picture than what we showed with our research. To this end, a follow-up study is required to explore the thresholds of translational gains fully.

Furthermore, a follow-up study of translational gains could also explore the effects of dynamic translational gains. Our usage of translation gains was purely static, and it would have been interesting to see how users would respond to a dynamic gain. As mentioned, a participant reported that they felt we were decreasing their speed during one of the curvature tests. This could allude to the fact that users can adapt to translation gains and adjust their physical walking speed to match the gain. If we were to test a dynamic gain, changes in the gain applied may have changed the thresholds as the user could not adapt to a single gain.

7.2.2 Acceptability Thresholds

In our study, we demonstrated that participants tend to find RDW acceptable even when the intensity of the gain is considered overt, as observed for both curvature and rotational gains. Despite several overt parameters, very few were deemed unacceptable. All the gains except the baseline were detected for rotational gains, but only the most intense gain was rated as unacceptable. Similarly, the two most intense gains were detected for curvature gains, but none were considered unacceptable. There was, however a clear downward trend for curvature gains, suggesting that more intense gains would likely have crossed the boundary to unacceptability.

Although we could not identify any statistical difference between the means of translational gains, the means are all above the acceptability threshold. Similar to detection, this may have been caused by the poorly selected parameters that did not reach the intensity where it became unacceptable. The slight downward trend in acceptability further supports this as the intensity grows. A follow-up study is necessary to fully explore the actual thresholds for the acceptability of translational gains.

As noted for detection thresholds and reported sickness, it is hard to compare the three techniques to each other as we can not be sure if the gains tested are translatable in terms of intensity (e.g., the highest gain for curvature, 5m, might not translate to the highest of rotation, 1.25). Additionally, as users generally reported that they found overt gains acceptable, they might affect the users' acceptability rating with more prolonged exposure to the overt gains. This, however, was not explored in our studies and requires a follow-up study that explores the effect of exposure time on acceptability.

7.2.3 Effects of Techniques

Our study observed no significant improvements in the number of resets, except for translation gains. Notably, translation gains demonstrated a strong increase in the distance walked between resets, while curvature gains showed a slight increase. However, for rotation gains, no consistent improvements in the distance between resets were found. Interestingly, the average distance to the center was lower for rotational gains, while the opposite trend was observed for curvature gains.

We speculate that the lower average distance to the center for rotational gains may suggest that the technique is effective but not efficient enough to prevent resets. A possible scenario is that the technique redirects users toward the center, but once they pass it, it fails to turn them around in time, resulting in a reset event. To improve the usability of rotational gains, lower gains could be employed. However, as noted in the general remarks, this would significantly increase the required testing time. Therefore, a follow-up study should investigate the effectiveness of this.

In the case of curvature gains, we assume that our RE was not large enough to accommodate a complete 360-degree orbit with our selected parameters as a radius, making it impossible to avoid resets in the VE designed for the technique. Had our RE been larger, it is conceivable that users could have experienced infinite walking along straight virtual paths. Nevertheless, we observe that as the parameter increases, there is an increase in the distance that users can walk before being reset, confirming that the technique is working, albeit not as effectively as needed to prevent resets entirely.

For translation gains, as we employed static gain for our test, it was expected that the distance a user could walk would increase as the intensity grew.

7.2.4 Participant Attributes

Although the main focus of this thesis lies in finding thresholds for RDW in a general population, we also investigate the various demographics within our participants. By exploring this, we hoped to uncover any biases that might be present and, if differences presented themselves, if we could quantify this into formal results. Among the participants that contributed to our tests, we grouped them by various attributes that they provided in the registration phase (Section 3.2.3). Running an analysis of variance on detection rate and acceptability, where results (Section 6.1.5) were grouped by participant group, we found several attributes that might have biased how users responded to RDW.

During internal testing and the pilot study, we noted that there might be a bias among people familiar with video games or VR experiences. Our results show that those proficient in either of the two[2], showed a threshold of perceptibility lower than less proficient participants. This was also the case for acceptability, where the proficient group rated all experiences with gains worse than less proficient participants.

We speculate that participants who often play games are more likely to adapt quickly to virtual settings, which may cause a higher sense of immersion. This increased immersion may enable a heightened awareness of anything that draws away from it, which overt RDW can do. However, to form any strong opinion on this, we need more insight into how the users perceived the experience, which we did not collect. Therefore, in any subsequent studies on RDW, it could be advantageous to collect more data on users' subjective experiences to uncover what might cause an increased sensitivity to RDW.

However, although the results show statistical significance, we cannot fully conclude that the results are valid. The problem stems from the fact that when using a Kruskal-Wallis test, the general guideline is to have at least five or more entries per grouping. In our studies, we, unfortunately, did not have a uniform distribution across the various groups, and thus some comparisons contained less than five participants. Therefore, although we still consider the results, we do not feel confident in concluding that the findings are correct. Therefore, we suggest that further studies explore our findings with larger and more diverse populations.

7.3 User Study 2

The second user study was focused around the usage of distractors within an immersive VE. The results discussed here can be found in section 6.2.

7.3.1 Effect on Detection and Acceptability

When looking at the results of detection rates and acceptability scores for our second user study, we could not find any conclusive evidence on the effect of distractors for RDW. When comparing the two scenarios, we only found a significant difference when comparing the baseline tests (without redirection) for redirection detection. From the means, we can see that users reported more false positives when distractors were present in the VE[3]. From this, it might be that the presence of distractors affects the experience; we can not say whether it affects redirection.

---

[2] There was a near 100% overlap between those that reported a high proficiency in both attributes.

[3] It is possible that these were, in fact, not false positives, as is discussed under *Remarks* (Section 7.3.2)

7.3.2 Effects of Distractors

Our results show that distractors had no significant effect on the distance traveled between resets in our experiments. Compared to what Chen and Fuchs [5] states in their findings, where they saw great success in redirection, we cannot confirm this. However, as has been remarked, we deemed our redirection controller unsuitable, or sub-optimal, for real-world application of RDW as it is too primitive to redirect in a way that limits resets effectively. Comparatively, Chen and Fuchs [5] used a predictive controller and a very controlled VE. In contrast, we employed a generalized controller in a VE where users had free reign to walk where they wanted. Therefore, although we would have hoped to match some of their findings, the two studies contain differences that set them apart.

However, even though we can not show a direct change in resets or distance between resets, we can show a significant increase in HMD rotation. While this does not affect the user's experience substantially, as it does not reduce resets, it is still a metric for the distractors' effects. From both observations during the tests, and the results presented, the increased HMD rotation allows the redirection technique to apply more rotation. An observation from the results is that the difference in added rotation between the parameters differs from close to 50% down to 13% (Table 6.9). The variety in these results surprised us, as we had no approach to speculate on why this happened. A possible reason could have been the number of times a participant encountered the wisp. However, due to lacking data collecting of this kind, we are unfortunately unable to investigate this.

Furthermore, had we had a steering algorithm that could align the user with their objective, we may have seen more effective use of the distractors? This, however, was not done in our study, and we consider this to be something that should be assessed for a follow-up study.

7.3.3 Remarks

The following are minor remarks that are specific to the second user study.

Cluttered with multiple distractors

During our testing, if the user engaged both the wizard and the wisp simultaneously, the user could sometimes appear overwhelmed. What was seen was that instead of being able to focus on a single distractor at a time, the attention of the user became split among the two. We do not consider this effect overly damning for the user's experience, but it did, at times, limit the usefulness of the wisp since the user could not actively follow it. Additionally, although no users reported this during the tests, it could have impacted some users' acceptability scores as they might not have enjoyed the somewhat overwhelming experience.

Improper baseline tests

In hindsight, we believe that using a *lower* gain during these tests was possibly not the optimal choice. The gain was used when testing all *upper* gains, including the base tests. The reasoning behind this choice was a hurried setup procedure where we wanted to see how our redirection controller could operate if given both an upper and lower gain. Ultimately, this did not lead to any visible benefits but instead made it so the base tests still contained redirection, which makes it hard for us to say if the reported detection during these tests were false positives or if the user noticed the small amount of redirection that was still being applied.

Range of interaction between participants

During the tests, some users became incredibly invested in the distractors and worked to defend themselves from the projectiles launched at them. In contrast, other participants did not pay much attention to the projectile mechanic. One participant even wholly ignored the distractors after a couple of tests as they realized there was no real penalty for being hit by the projectiles. After the study, this participant stated that they did not *care* about the gameplay aspect and did not feel invested to any degree. In this instance, and partly related to other participants, it may have been beneficial to add a greater incentive or a penalty for being hit that further gave reason to interact with the distractors. Alternatively, we could have instructed the users to interact with the distractors. However, at the time of the study design and during the study, we did not feel this to be a good choice as it might reduce the sense of presence users felt if they were *made* to do something.

7.4 Summary

7.4.1 Research Question 1

In terms of acceptability, we are able to show confident results for both rotational and curvature gains, while translation gains remain inconclusive. We can show that users will generally deem redirection acceptable, even if it is overt. For curvature gains, we show that users find it acceptable with gains as intense as 5.0 meters, possibly even stronger. Rotational gains show a similar trend where users report that gains as strong as 1.4 are acceptable, with 1.55 being just past the threshold of unacceptability. Additionally, although not considered confident enough, we note a trend that shows users with higher proficiency in video games and VR rate experiences are less acceptable than those who are less proficient.

7.4.2 Research Question 2

We were able to show that there is a growing trend in perceptibility for rotational and curvature gains, while translational gains remain inconclusive. For rotational gains, we show that users are able to detect gains as low as 1.1 and that the threshold lies between 1.0 and 1.1. For curvature gains, we show that users are able to perceive curvature gains when the radius becomes lower than 10.0 meters. Although we could not say anything confident about translation gains, the threshold appears to lie at, or just above, our most intense parameter, 1.25. To explore this threshold, a follow-up study would be beneficial. As n the case of acceptability, those with higher proficiency in video games and VR rate experiences are able to detect gains sooner than those who are less proficient.

7.4.3 Research Question 3

During our distractor trials, we could not confirm if having distractors present in the VE changed the perception thresholds of users. However, looking at the data, there appears to be a possibility that distractors extend the range of where gains remain imperceptible, but this cannot be concluded in this study.

7.4.4 Research Question 4

With clear evidence, we are able to show that by integrating a distractor into the VE, the amount of HMD rotation a user will perform is greatly increased. Although we could not show any clear decrease in resets or other measurable benefits, we show that the increase in HMD rotation gives the redirection technique more opportunities to apply reorientation. Our results show that with our distractor, it is possible to increase the amount of HMD rotation between 13% and 43% compared to a scenario without the distractors.

Chapter 8

Conclusion

In this thesis, the primary objective was to investigate Redirected Walking in Virtual Reality (VR). The focal point of our research revolved around two aspects: first, identifying the thresholds where users perceive various redirection techniques as unacceptable, and second, evaluating the impact of incorporating distractors in the VR scenario on user experience and its effect on redirection. Our motivation stemmed from inconsistencies in threshold values and effects observed in previous research. Moreover, we identified that literature addressing the influence of distractors on VR experiences was few and far between, warranting further investigation.

We evaluated three prevalent redirection techniques: *Curvature Gains*, *Rotational Gains*, and *Translational Gains*. To assess the effects of distractors, we also tested rotational gains with distractors present. These techniques were examined through a user study, using our implementation of the various redirected walking techniques.

The results of the user studies show that users are generally more likely to detect the presence of redirection as the intensity of the gains grows. For rotational gains, we were able to show that users are able to detect amplification as low as 10%, while the thresholds for curvature appeared at 7.5m. Due to low confidence in results for translational gains, we can not determine a threshold.

Similarly, the acceptance of RDW diminishes as the level of gain increases. However, our findings indicate that users generally find gains acceptable, even if they detect that redirection is happening. For rotation gains, a threshold appears between 40% and 55% amplification in angular velocity, where users report an unacceptable experience. A similar trend occurs with curvature gains - as the gain increases, the level of user acceptability decreases. While users were more accepting of larger curvature radii, they still found radii as low as $5m$ to be acceptable. Though we did not identify a specific threshold for translational gain acceptability, we observed that for all the different gain intensities tested, as high as an amplification of 25% was considered acceptable for users.

To evaluate distractors, we conducted a second user study using only rotational gains, where users were placed in a game-like scenario featuring monsters/distractors. Although we cannot definitively conclude that distractors directly affected users' awareness or acceptability of gains, our results indicate that the presence of distractors strongly influences the effectiveness of the rotational gain technique. With distractors present, we showed that the average head rotation, a key element for rotational gain, increased by 13% to 43%.

In conclusion, this thesis delves into the exploration and evaluation of thresholds for various redirection techniques and examines the potential effects of incorporating distractors in the VR experience.

8.1 Future Work

This section will discuss potential directions for further research and improvements of the concepts presented in my thesis. Chapter 7 has already discussed some future work.

- **Additional Redirection Techniques:** In addition to the three redirection techniques considered for this thesis, there exist others [29]. Similarly to how we considered our techniques to have room for further explorations, some techniques are even less explored and could benefit from additional research.
- **Larger Scale Study:** For many of the places where we were not able to conclude with statistical significance, we might have been able to solve it with a larger and more diverse population of participants. Additionally, exploring a broader, more granular range of gains would expand the research presented in this thesis and possibly uncover what we could not (e.g., thresholds for translation gain). Lastly, we noted in section 7.2.3 that our usage of curvature gains did not have enough room to function optimally; a further study with a larger RE could explore this.
- **Better Controller:** Although not the key focus of this thesis, as discussed in section 7.1, what we consider the key reason why we did not see substantial benefits from RDW in our studies was due to an ill-fitted redirection controller. In further studies, we suggest the implementation of a redirection controller that can redirect the user in a way that properly limits reset events.
- **Smarter Distractors:** A concept that we considered early in development was the use of *smart* distractors. In contrast to our distractors, these distractors could dynamically move around the user, or VE, so that they move in specific directions. For our thesis, we considered this to be overly advanced, but a follow-up study exploring this could possibly extend the usability of distractors in RDW.

Appendix A

Participant Instruction Pamphlet

RDW Test Candidate Notes

After reading this small document you will hopefully be familiar with the various tests that you will go through in this study. It is not necessary to know everything by heart, but please get somewhat familiar with how the tests work and what questions you will answer between tests.

Important notes

- There are mechanisms in place to prevent you from walking into obstacles. There will also be a person around to stop you in case this fails. Still we advise that you never move faster than a brisk walk.
    - For the stopping mechanism to function, you will personally have to respond to the prompt. If you are moving too fast you may not have time to physically stop within 1-2 meters.
    - The same goes for the person who will oversee you. If you start moving at a fast pace then the person may not be able to respond in time.
- If at any time during the test you feel physically unwell or feel that the test is too uncomfortable to continue. Please tell your test conductor and they will skip the current test.
    - You may also at any time stop the test completely if you do not wish to
- Technical difficulties
    - If an issue with the VR headset should occur, simply tell your conductor and they will hopefully be able to solve the problem
    - This also goes if you are experiencing any “lag” or stutters during your test

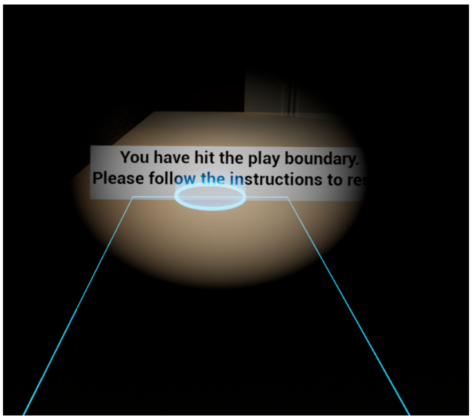
“Resetting”

Resetting will be a common occurrence during your tests. A reset is simply a mechanism to stop you from going out of bounds (crashing into a wall). The reset mechanism is not meant to be a key part of the test, and should not be considered “reorientation” for the questionnaire later on, it is simply a necessary mechanism to allow for simple continuation upon hitting a boundary.


You have hit the play boundar
Please follow the instructions to r
Please perform a 360 degree turn
nd align yourself to face the blue c

A reset will do the following upon detecting that you are on a boundary:

1. Enable “tunnel vision” (dim the screen)
2. Show a pop-up (“You have hit a boundary”)
3. Instruct you to perform a rotation to line up with a blue circle
    a. The virtual world will be rotate 360 degrees so that once you have lined up with the circle, you are facing the same direction in VR as before the reset
    b. To avoid motion sickness during this phase, focus only on the blue overlaid elements and not the world in the background
4. Once lined up with the circle a blue “pathway” will show up. To complete the reset, follow this path towards the blue circle
    a. You will have to walk approximately 1.5 meters forward
5. Once you have walked far enough forward, the pop-up will disappear and the tunnel vision will fade
6. You can now continue the test


You have hit the play boundary.
Please follow the instructions to re

Tutorial/Start Room

This is the first room you will enter, in this room you will see a screen that states your test ID number and one button.

If this is your first time in this room please:

1. Press the **RED** button
    a. This will enable resetting
2. Walk slowly into the translucent gray wall
    a. This should trigger a reset
    b. For it to trigger your head must fully pass through the wall
3. Follow the instructions and get familiar with how resetting works

    a. Please try this a couple of times so that you know what to do once prompted during real tests
4. Once familiar, tell the person conducting the test that you are ready to move on.

Intermission Room

This is the second room that you will enter. In this room you will find one button. Pressing this button will take you to the next test that will be conducted.

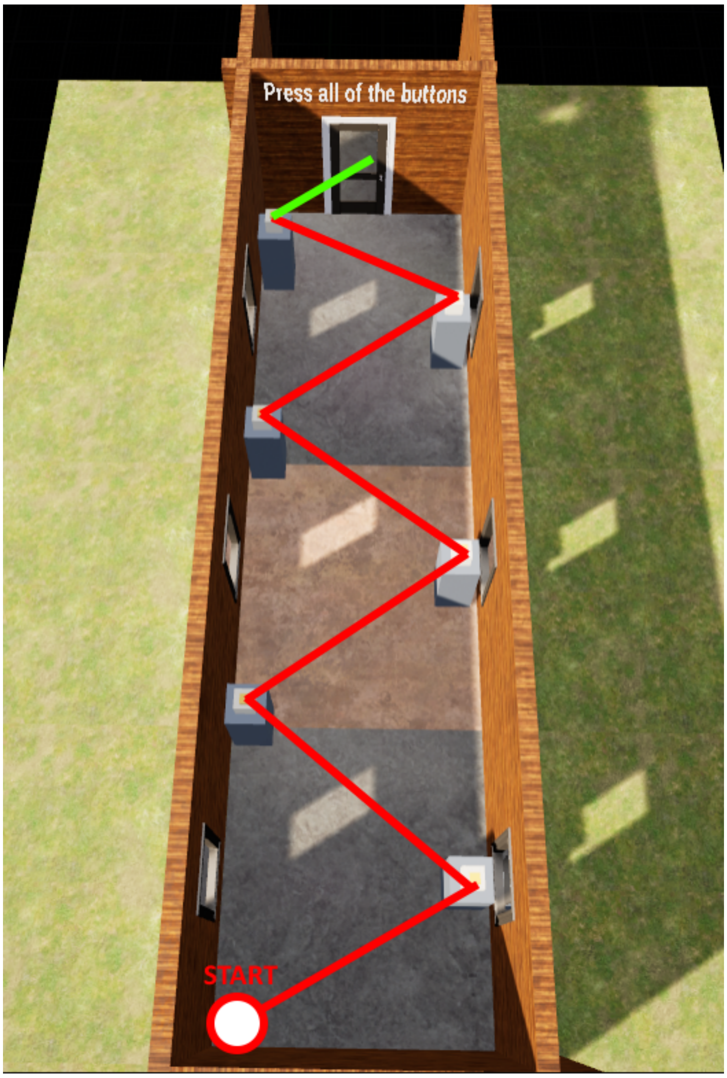
If you have just completed a test you will also be transported to this room. Before moving onto the next test you will first answer the following questions about your previous tests:

1. Did you notice any redirection/reorientation in your previous test?
    a. Yes / No answer
    b. Reorientation can be explained as unnatural rotation/movement during the test that was not caused by you
        i. Example: A noticeable rotation of your view
2. How acceptable was the experience:
    a. On a scale from 1-5
        i. 1 is completely unacceptable and 5 is perfectly acceptable
3. How “sick” would you rate yourself right now?
    a. On a scale from 1-5
        i. 1 is not sick at all, 5 is very sick (i.e. about to vomit)
4. (Only if you reported being sick) Please describe how you feel.
    a. What symptoms led you to answering that you felt some degree of being “sick”
    b. Nausea, dizziness, headache etc.
    c. The test conductor will mention some symptoms for you to agree/disagree with

Tests

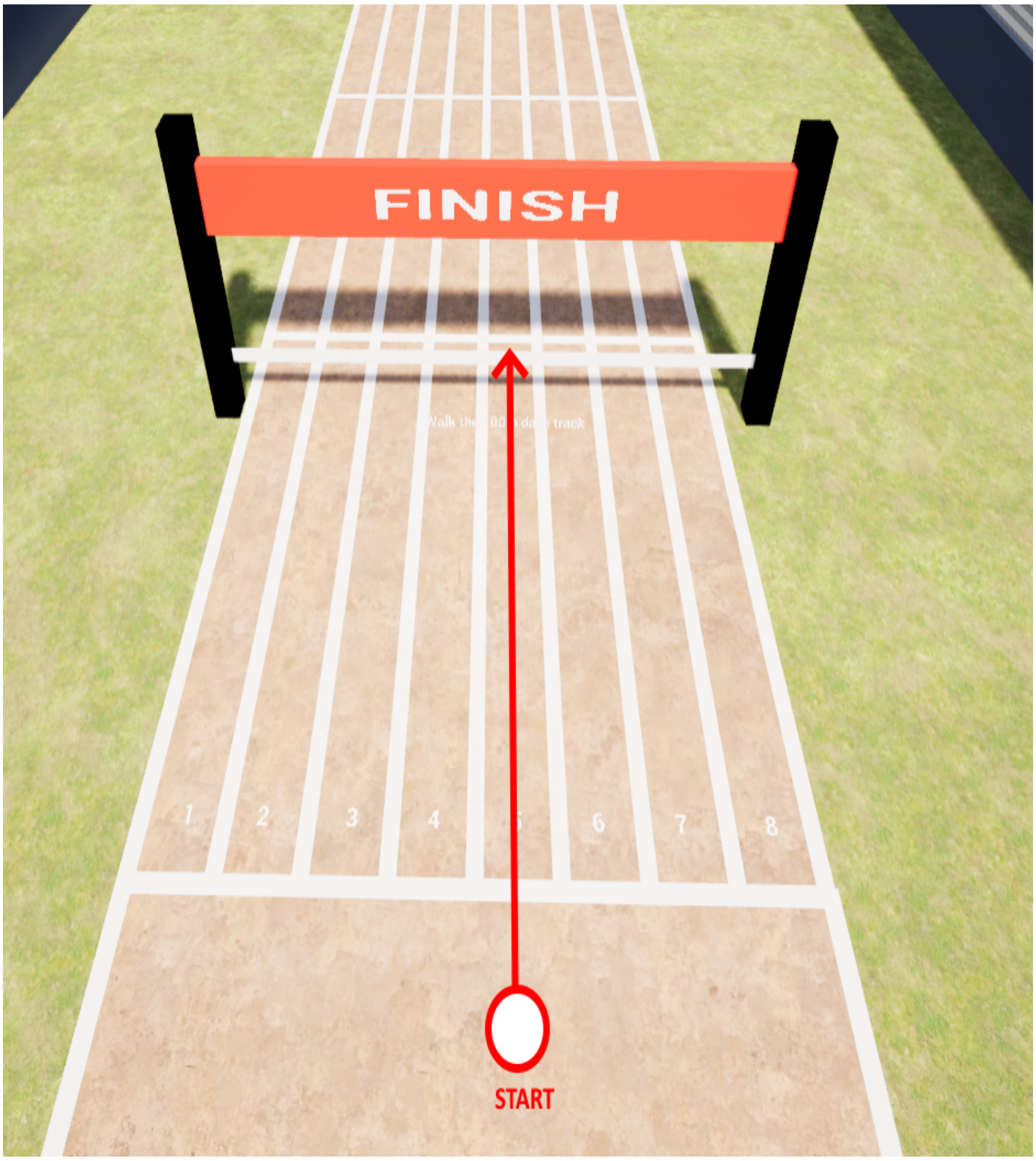
Zig-Zag

In this test your goal is to move between all buttons in the level in a Zig-Zag pattern. Once a button has been pressed, it will turn green. Once it has turned green you may move to the next button. To exit the level, walk through the door once it opens.



Track and field

In this test you will be transported to a running circuit. Your goal is simply to walk to the finish line. (When you enter the level the goal line might be behind you)



Witch’s Cauldron

In this level your task is to locate the correctly colored ball (which is shown above the cauldron) and place it in the cauldron. A new color will then appear and you will have to do the same. This is done four times before the test is done.

Welcome to the potion test
Look at the rotating color sphere over the cauldron.
Grab the correct sphere, and drop it in the cauldron
If it is the wrong color, it will be thrown out of the cauldron
Get all the colors in to pass the test

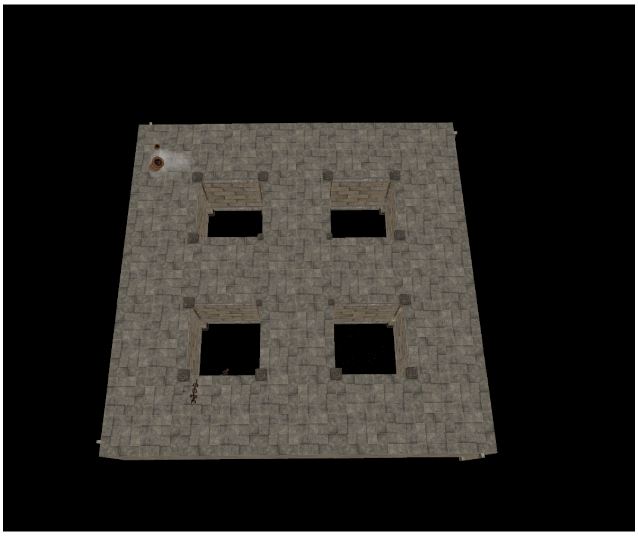
Velkommen til heksegryta
Se på kula i midten
Finn den riktige fargen, og slipp den i gryta
Hvis du velger feil farge, vil den bli kastet ut av gryta
Plasser de riktige fargene for å bestå

Tests

The Dungeon

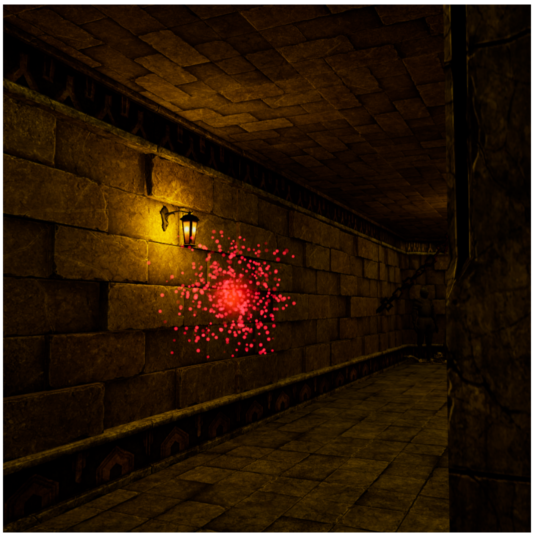
In the dungeon your task is to find all the gold and riches, but beware! There might be evil creatures wanting to claim it for themselves!

1. The Dungeon has a square layout, with a central crossroads. See Figure:



2. Your Task is to locate and open 3 chests. The chests are filled with gold that you should collect. Chests are opened by simply touching the lock. You may then collect the gold by touching it.

3. During your tests, half of them will contain “NPCs” while the other half of tests will be void of them. Each NPC has their own logic.
    a. “The wisp” is a creature that will appear from any direction of the map at random times. The wisp will approach you and start circling you. At a random time during its orbit it will fire its “spell”. Using your shield you must block the spell to avoid losing your precious gold



    b. “The Wizard” is a roaming creature that will walk around the dungeon. If it spots you it will first approach, before then attacking. Similar to the wisp, the Wizard will launch a “spell” which you must block using your shield. The unique part of the Wizard is that all spells blocked will be launched back at the wizard. If the wizard takes damage twice it will perish.

If a reset occurs while engaging any of the NPCs their logic will be paused, and any spells in the air will disappear.

Bibliography

[1] 'A Steering Algorithm for Redirected Walking Using Reinforcement Learning'. In: *IEEE Transactions on Visualization and Computer Graphics* 26 (5 May 2020), pp. 1955–1963. ISSN: 1077-2626. DOI: 10.1109/TVCG.2020.2973060. URL: https://ieeexplore.ieee.org/document/8998570/.

[2] Costas Boletsis. 'The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology'. In: *Multimodal Technologies and Interaction* 1.4 (2017). ISSN: 2414-4088. DOI: 10.3390/mti1040024. URL: https://www.mdpi.com/2414-4088/1/4/24.

[3] G. Bruder et al. 'Redirecting Walking and Driving for Natural Navigation in Immersive Virtual Environments'. In: *IEEE Transactions on Visualization and Computer Graphics* 18 (4 Apr. 2012), pp. 538–545. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.55.

[4] Yang Hun Cha, Dang Yang Lee and In Kwan Lee. 'Path Prediction Using LSTM Network for Redirected Walking'. In: Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 527–528. ISBN: 9781538633656. DOI: 10.1109/VR.2018.8446442.

[5] Haiwei Chen and Henry Fuchs. 'Towards imperceptible redirected walking: Integrating a distractor into the immersive experience'. In: 2017. DOI: 10.1145/3023368.3036844.

[6] Bethany Cieslowski et al. 'The Development and Pilot Testing of Immersive Virtual Reality Simulation Training for Prelicensure Nursing Students: A Quasi-Experimental Study'. In: *Clinical Simulation in Nursing* 77 (2023), pp. 6–12. ISSN: 1876-1399. DOI: https://doi.org/10.1016/j.ecns.2023.02.001. URL: https://www.sciencedirect.com/science/article/pii/S1876139923000087.

[7] Ajoy S Fernandes and Steven K. Feiner. 'Combating VR sickness through subtle dynamic field-of-view modification'. In: IEEE, Mar. 2016, pp. 201–210. ISBN: 978-1-5090-0842-1. DOI: 10.1109/3DUI.2016.7460053.

[8] *File:BeepMD JP 1994-08.pdf - Sega Retro*. [Online; accessed 24. Apr. 2023]. Apr. 2023. URL: https://segaretro.org/index.php?title=File:BeepMD_JP_1994-08.pdf&page=31 (visited on 24/04/2023).

[9] Cynthia L. Foronda et al. 'Virtual Simulation in Nursing Education: A Systematic Review Spanning 1996 to 2018'. In: *Simulation in Healthcare* 15.1 (Feb. 2020), p. 46. ISSN: 1559-2332. DOI: 10.1097/SIH.0000000000000411. (Visited on 24/04/2023).

[10] Timofey Grechkin et al. 'Revisiting detection thresholds for redirected walking'. In: ACM, July 2016, pp. 113–120. ISBN: 9781450343831. DOI: 10.1145/2931002.2931018.

[11] Heather. 'Oculus: Complete Guide — History, Products, Founding, and More'. In: *History-Computer* (Dec. 2022). URL: https://history-computer.com/oculus-history (visited on 24/04/2023).

[12] *History Of Virtual Reality - Virtual Reality Society*. [Online; accessed 24. Apr. 2023]. Jan. 2020. URL: https://www.vrs.org.uk/virtual-reality/history.html (visited on 24/04/2023).

[13] Eric Hodgson and Eric Bachmann. 'Comparing Four Approaches to Generalized Redirected Walking: Simulation and Live User Data'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.4 (2013), pp. 634–643. DOI: 10.1109/TVCG.2013.28.

[14] KATVR. *Kat Walk C2*. 2022. URL: https://www.kat-vr.com/products/kat-walk-c2 (visited on 22/05/2022).

[15] 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.

[16] Lucie Kruse, Eike Langbehn and Frank Steinicke. 'I Can See on My Feet While Walking: Sensitivity to Translation Gains with Visible Feet'. In: IEEE, Mar. 2018, pp. 305–312. ISBN: 978-1-5386-3365-6. DOI: 10.1109/VR.2018.8446216.

[17] Eike Langbehn, Paul Lubos and Frank Steinicke. 'Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking'. In: *Proceedings of the Virtual Reality International Conference - Laval Virtual*. VRIC '18. Laval, France: Association for Computing Machinery, 2018. ISBN: 9781450353816. DOI: 10.1145/3234253.3234291. URL: https://doi.org/10.1145/3234253.3234291.

[18] Eike Langbehn and Frank Steinicke. 'Redirected Walking in Virtual Reality'. In: Springer Encyclopedia of Computer Graphics and Games, 2018, pp. 1–11. URL: http://basilic.informatik.uni-hamburg.de/Publications/2018/LS18.

[19] Eike Langbehn et al. 'Bending the Curve: Sensitivity to Bending of Curved Paths and Application in Room-Scale VR'. In: *IEEE Transactions on Visualization and Computer Graphics* 23 (4 Apr. 2017), pp. 1389–1398. ISSN: 1077-2626. DOI: 10.1109/TVCG.2017.2657220.

[20] Eike Langbehn et al. 'In the blink of an eye - Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality'. In: *ACM Transactions on Graphics* 37 (4 2018). ISSN: 15577368. DOI: 10.1145/3197517.3201335.

[21] 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.

[22] Yi-Jun Li et al. 'OpenRDW: A Redirected Walking Library and Benchmark with Multi-User, Learning-based Functionalities and State-of-the-art Algorithms'. In: *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2021, pp. 21–30. DOI: 10.1109/ISMAR52148.2021.00016.

[23] Yi Jun Li, Frank Steinicke and Miao Wang. 'A Comprehensive Review of Redirected Walking Techniques: Taxonomy, Methods, and Future Directions'. In: *Journal of Computer Science and Technology* 37 (3 June 2022), pp. 561–583. ISSN: 18604749. DOI: 10.1007/s11390-022-2266-7.

[24] James Jeng-Weei Lin et al. 'Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment'. In: *Proceedings IEEE Virtual Reality 2002* (2002), pp. 164–171.

[25] H. Mayo. *Outlines of Human Physiology*. Burgess and Hill, 1833. URL: https://books.google.no/books?id=RawLAQAAIAAJ&q=Herbert+Mayo+1833&pg=PA288&redir_esc=y#v=onepage&q&f=false (visited on 24/04/2023).

[26] Ronald R. Mourant and Thara R. Thattacherry. 'Simulator Sickness in a Virtual Environments Driving Simulator'. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 44 (5 July 2000), pp. 534–537. ISSN: 2169-5067. DOI: 10.1177/154193120004400513.

[27] Thomas Nescher, Ying-Yin Huang and Andreas Kunz. 'Planning redirection techniques for optimal free walking experience using model predictive control'. In: IEEE, Mar. 2014, pp. 111–118. ISBN: 978-1-4799-3624-3. DOI: 10.1109/3DUI.2014.6798851.

[28] Lasse T. Nielsen et al. 'Missing the point'. In: ACM, Nov. 2016, pp. 229–232. ISBN: 9781450344913. DOI: 10.1145/2993369.2993405.

[29] Niels Christian Nilsson et al. '15 Years of Research on Redirected Walking in Immersive Virtual Environments'. In: *IEEE Computer Graphics and Applications* 38 (2 Mar. 2018), pp. 44–56. ISSN: 0272-1716. DOI: 10.1109/MCG.2018.111125628. URL: https://ieeexplore.ieee.org/document/8255772/.

[30] *OpenXR - High-performance access to AR and VR —collectively known as XR— platforms and devices*. [Online; accessed 5. May 2023]. Dec. 2016. URL: https://www.khronos.org/openxr (visited on 05/05/2023).

[31] Anders Paludan et al. 'Disguising Rotational Gain for Redirected Walking in Virtual Reality: Effect of Visual Density'. In: (Jan. 2016).

[32] Tabitha C. Peck, Henry Fuchs and Mary C. Whitton. 'Evaluation of Reorientation Techniques and Distractors for Walking in Large Virtual Environments'. In: *IEEE Transactions on Visualization and Computer Graphics* 15.3 (2009), pp. 383–394. DOI: 10.1109/TVCG.2008.191.

[33] Tabitha C. Peck, Henry Fuchs and Mary C. Whitton. 'Improved Redirection with Distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments'. In: IEEE, Mar. 2010, pp. 35–38. ISBN: 978-1-4244-6237-7. DOI: 10.1109/VR.2010.5444816.

[34] Sharif Razzaque, Zachariah Kohn and Mary C. Whitton. 'Redirected Walking'. In: *Eurographics 2001 - Short Presentations*. Eurographics Association, 2001. DOI: 10.2312/egs.20011036.

[35] Hiroaki Sakono et al. 'Redirected Walking using Continuous Curvature Manipulation'. In: *IEEE Transactions on Visualization and Computer Graphics* 27 (11 Nov. 2021), pp. 4278–4288. ISSN: 19410506. DOI: 10.1109/TVCG.2021.3106501.

[36] *Steam Hardware & Software Survey*. [Online; accessed 24. Apr. 2023]. Apr. 2023. URL: https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam (visited on 24/04/2023).

[37] F. Steinicke et al. 'Estimation of Detection Thresholds for Redirected Walking Techniques'. In: *IEEE Transactions on Visualization and Computer Graphics* 16 (1 Jan. 2010), pp. 17–27. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.62.

[38] Frank Steinicke et al. 'Analyses of human sensitivity to redirected walking'. In: ACM, Oct. 2008, pp. 149–156. ISBN: 9781595939517. DOI: 10.1145/1450579.1450611.

[39] Frank Steinicke et al. 'Moving Towards Generally Applicable Redirected Walking'. In: *Proceedings of IEEE VRIC 2008 : 10th International Conference on Virtual Reality* (Jan. 2008).

[40] Evan A. Suma et al. 'Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture'. In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (2012), pp. 555–564. DOI: 10.1109/TVCG.2012.47.

[41] Evan A. Suma et al. 'Leveraging change blindness for redirection in virtual environments'. In: *2011 IEEE Virtual Reality Conference*. 2011, pp. 159–166. DOI: 10.1109/VR.2011.5759455.

[42] Evan Suma Rosenberg et al. 'A taxonomy for deploying redirection techniques in immersive virtual environments'. In: *Proceedings - IEEE Virtual Reality* (Mar. 2012), pp. 43–46. DOI: 10.1109/VR.2012.6180877.

[43] Jerald Thomas and Evan Suma Rosenberg. 'A general reactive algorithm for redirected walking using artificial potential functions'. In: 2019. DOI: 10.1109/VR.2019.8797983.

[44] Usc-Ict-Mxr. *RDWT*. [Online; accessed 5. May 2023]. May 2023. URL: https://github.com/USC-ICT-MxR/RDWT (visited on 05/05/2023).

[45] Virtuix. *Omni by Virtuix*. 2022. URL: https://www.virtuix.com/ (visited on 22/05/2022).

[46] 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.

[47] Betsy Williams et al. 'Updating Orientation in Large Virtual Environments Using Scaled Translational Gain'. In: APGV '06. Boston, Massachusetts, USA: Association for Computing Machinery, 2006, pp. 21–28. ISBN: 1595934294. DOI: 10.1145/1140491.1140495. URL: https://doi.org/10.1145/1140491.1140495.

[48] M. A. Zmuda et al. 'Optimizing Constrained-Environment Redirected Walking Instructions Using Search Techniques'. In: *IEEE Transactions on Visualization and Computer Graphics* 19 (11 Nov. 2013), pp. 1872–1884. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.88.