FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Navigation Techniques for Seated VR

Guilherme dos Santos Amaro

DISSERTAÇÃO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Daniel Filipe Martins Tavares Mendes

Navigation Techniques for Seated VR

Guilherme dos Santos Amaro

Mestrado Integrado em Engenharia Informática e Computação

Abstract

Virtual reality (VR) has recently seen a greater interest by both consumers and enterprises due to its capabilities in simulation and immersion, as well as information display and interaction methods; the latter are often associated with the use of natural user interfaces (NUI). However, typical VR interactions can be tiring, including standing up, walking, and mid-air gestures. These can reduce comfort and session duration compared with an equivalent non-VR interface, which may, in turn, lead to decreased productivity. Nevertheless, current approaches often neglect this aspect, making the VR experience not as promising as it can be.

We believe that seated VR experiences can be a good compromise between overall experience and ergonomics, allowing for the convenience and comfort of a desktop experience and the benefits of VR immersion.

In this work, we explore navigation techniques adequate for seated VR users. We target travel and orientation techniques separately, considering only continuous approaches for travel, as these are more adequate for environment exploration. We consider both continuous and discrete approaches for orientation.

Results from this dissertation should benefit a large spectrum of users. People with physical disabilities cannot stand for long periods of time and would no longer be prevented from experiencing VR comfortably. Professionals who make use of VR can increase their daily productivity through longer sessions without the need for a break. Videogame players would benefit from extended playtime, making the system more cost-effective.

With this in mind, results show that our set of techniques have proven to be effective at traversal of a virtual environment. We also found possible trends towards task focus affecting overall symptomatology, as well as a preference for controller-based methods for comfort and ease-of-use.

ACM Classification: CCS → Human-centered computing → Human computer interaction (HCI) → Interaction techniques

Keywords: virtual environment, human computer interaction, interaction technique, navigation technique, gestures

Resumo

Realidade virtual (VR) tem visto um grande aumento de interesse por parte de tanto consumidores como empresas. Isto deve-se às suas capacidades de simulação e imersão, bem como na visualização de informação e métodos de interação; Estes últimos encontram-se frequentemente associados com o uso de interfaces naturais de utilizador (NUI). No entanto, navegação e interação em VR podem ser cansativas devido às mesmas, incluindo ficar de pé, andar, correr, saltar e gesticular no ar. Isto pode reduzir o conforto e duração das sessões quando comparado com um equivalente não-VR, o que, consequentemente, leva a redução de produtividade. Ainda assim, abordagens actuais frequentemente negligenciam este aspecto, tornando a experiência em VR menos promissora do que poderia ser.

Para este fim, o principal foco desta dissertação será propor uma alternativa às típicas NUI com um método adequado de navegação para VR no formato sentado que use gestos e controlo táctil. VR sentado tira partido da conveniência e conforto de uma experiência de uma mesa de trabalho, enquanto que os gestos e controlo tactíl agem como NUI alternativas, chegando a um bom compromisso entre experiência em geral e ergonomia.

O resultado desta dissertação deverá beneficiar um grande espectro de utilizadores. Pessoas com deficiências físicas não conseguem manter-se de pé por períodos longos de tempo, passando assim a conseguir experienciar VR confortávelmente. Profissionais que usam VR conseguiriam aumentar a sua productividade diária com sessões prolongadas, sem necessidade de pausas. Videojogadores também beneficiariam de mais tempo de jogo, tornando os sistemas mais rentáveis.

Com isto em mente, os nossos resultados mostram que o nosso conjunto de técnicas provou ser effectivo como método de navegação de uma ambiente virtual. Também encontramos possíveis tendências de que o foco numa tarefa pode afectar sintomatologia em geral, bem como uma preferência a nível de conforto e facilidade de acesso de métodos que usam comandos VR.

Classificação ACM: CCS $\rightarrow$ Human-centered computing $\rightarrow$ Human computer interaction (HCI) $\rightarrow$ Interaction techniques

Palavras-chave: ambiente virtual, interfaces computador-humano, técnicas de interação, técnicas de navegação, gestos

Acknowledgements

This project was financed by ERDF - European Regional Development Fund through Programa Operacional para a Competitividade e Internacionalização - COMPETE 2020 within the scope of Portugal 2020 Partnership Agreement, and by national funds through the portuguese financing agency FCT - Fundação para a Ciência e a Tecnologia within the PAINTER project with reference POCI-01-0145-FEDER-030740 - PTDC/CCI-COM/30740/2017.

I would like to thank my dissertation proponent and supervisor Daniel Mendes for all the help and support given during this past year. Not just for submitting this dissertation theme, nor for allowing me to work on it, but for his continued assistance and supervision which, with the pandemic in full force, the workload from all my other classes, and personal life complications, still kept me on the right track.

I also want to thank my mother and my friends for all of their daily support. This has been a very hard year for everyone, emotionally and otherwise.

"Life is a journey, not a destination."

???

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Problem 1
1.3 Objective 2
1.4 Contributions 2
1.5 Publication 2
1.6 Document structure 3

2 State of the Art **5**
2.1 Current navigation techniques 5
2.1.1 Travel 5
2.1.2 Orientation 8
2.2 Presence and cybersickness 9
2.3 Ergonomics 9
2.4 Discussion 10

3 Proposed techniques **13**
3.1 Locomotion 13
3.1.1 Continuous Directional Movement 13
3.1.2 *Dog Paddle* 14
3.1.3 Drag'n'Go 14
3.2 Orientation 15
3.2.1 Continuous Directional Rotation 15
3.2.2 Tactile Surface Dragging 16
3.2.3 *Choose & Click* 16
3.2.4 Gaze Convergence 17
3.3 Summary 17

4 Prototype **19**
4.1 Hardware Setup 19
4.2 Architecture 19
4.3 Virtual Testing Environment 20
4.4 VR Optimization 21
4.5 Development Notes 23
4.6 Summary 24

5 Evaluation **25**
5.1 Methodology 25
5.2 Tasks 26
5.2.1 Travel 26
5.2.2 Orientation 28
5.3 Participants 29
5.4 Summary 29

6 Results **31**
6.1 User Performance 31
6.2 Questionnaire Results 33
6.3 Observations 38
6.4 Discussion 39
6.5 Summary 41

7 Conclusions **43**
7.1 Future Work 43

References **45**

List of Figures

2.1 Common movement technique examples[1]. *Teleport* is a discrete method of movement, while *Continuous* is as its name implies. 6
2.2 Magic Carpet example. From [16]. 7
2.3 *Drag'n'Go* method[17] example 8
2.4 Example of postures during interaction with surfaces[2] 10
2.5 *Drag'n'Go* example in VR 12

3.1 Continuous Directional Movement locomotion technique 13
3.2 *Dog Paddle* locomotion technique 14
3.3 *Drag'n Go* locomotion technique 15
3.4 Continuous Directional Rotation technique 15
3.5 Tactile Surface Dragging technique 16
3.6 *Choose & Click* technique 16
3.7 Gaze Convergence technique 17

4.1 Architecture diagram 20
4.2 Final version of the Virtual Testing Environment 21
4.3 Example of the shader used for buildings 22
4.4 Examples of performance graph available through SteamVR. Left image shows an early version of the prototype with stable performance, albeit with a few dropped frames; Right image shows an example of poor performance[2]. Green means no issues, yellow means a previous frame was re-projected for taking too long to render, orange means the frame was rendered late without re-projection, and red means the frame was skipped entirely, resulting in very noticeable stutter. 23

5.1 *Travel* task example. The heads-up display shows the current checkpoint number, with the arrow on top pointing towards it. 26
5.2 Mixed *Travel* task, starting on the right-most checkpoint 27
5.3 Horizontal *Travel* task, starting on the right-most checkpoint 27
5.4 Vertical *Travel* task, starting on the right-most checkpoint 28
5.5 *Orientation* checkpoint (right), *Orientation* task example (left) 29
5.6 Participants using VR controller (left) and touch surface (right) 29

6.1 Symptomatology for *Travel* tasks 36
6.2 Symptomatology for *Orientation* tasks 37
6.3 Example of participant only using small portion of available space 40

[2]fps

List of Tables

2.1 Movement methods 11

6.1 *Travel* technique times, [a] [b] [c] mean pair of statistic significance. 31
6.2 *Travel* technique measurements, [a] [b] [c] mean pair of statistic significance. 32
6.3 *Orientation* technique times, [a] [b] [c] mean pair of statistic significance. 32
6.4 *Orientation* technique measurements, [a] mean pair of statistic significance. 32
6.5 Performance Self-assessment for *Travel* techniques, [a] and [b] mean pair of statistic significance. 33
6.6 Travel technique assessment, [a] mean pair of statistic significance. 34
6.7 Performance Self-assessment for *Orientation* techniques, [a] [b] [c] mean pair of statistic significance. 34
6.8 *Orientation* technique assessment, [a] [b] mean pair of statistic significance. 35

Abbreviations

VR	Virtual Reality
VE	Virtual Environment
HCI	Human-computer interface
NUI	Natural User Interfaces
HMD	Head-mounted Display
FWIP	Finger Walking-In-Place
CDM	Continuous Directional Movement
DnG	Drag'n'Go
DP	Dog Paddle
CDR	Continuous Directional Rotation
C&C	Choose & Click
GC	Gaze Convergence
TSD	Tactile Surface Dragging
IQR	Interquartile Range

Chapter 1

Introduction

VR has been studied and developed over many years. However, only in the past decade has the technology become feasible or affordable for mass consumption. As such, a large amount of documentation only recently surfaced following the increase in relevance.

1.1 Motivation

Virtual Reality (VR) has always been an area of great interest due to its premise: the exploration and manipulation of a fictional, virtual environment (VE), from 3D imagery visualization to large scenes. These environments often come in varied sizes, ranging from a small room to an open world. While natural metaphors like walking can be desirable, they are often limited by hardware and available physical space. They may also lead to fatigue, reducing user performance. As such, the experience might become bound to a seated one, but even so, locomotion in the VE may still be required. Currently, some methods of navigation already exist that can be applied to seated VR. However, most were not made for that use case. Additionally, conclusions may not be as clear as one would hope. This is due to each method being compared with only a few others and testing methodology changing from evaluation to evaluation.

1.2 Problem

One of the biggest questions when talking about VR in general is how to interact and explore the VE in which the user is placed. Nowadays, even consumer-grade VR headsets often feature some type of simple tracking. If controllers are included, this means both head and hands can be tracked through both outside-in and inside-out methods. This information can then translated into user movement, for example. However, in seated VR, those points of reference become grounded in a chair, and so does the user, virtually.

On the flip side, another big issue is cybersickness. In particular, when movement in the VE does not match real life, users tend to experience motion sickness, nausea, and many other symptoms. By definition, **seated** *movement* causes a conflict of sensorial input.

With this in mind, **how can users retain an immersive VR experience while seated, without sacrificing physical comfort or wellbeing?** At the same time, we can start looking for directions by asking **which implementations work best for seated VR, and most importantly, why.**

1.3 Objective

We are looking for a movement method that reduces the compromise between sickness, fatigue, and presence. To this end, there will be a focus on VR devices that make use of natural touch and gesture metaphors. However, currently available methods still prove insufficient and do not always satisfy every parameter.

This project aims to propose navigation methods to allow natural and efficient exploration of a virtual environment, to be applied to a seated experience. This will include state-of-the-art implementations not necessarily made with a seated experience in mind and a new method that we will propose. **It is also our goal to ascertain aspects and features of those methods that contribute to a good experience, and those that negatively impact it.**

1.4 Contributions

The main contributions of this project are as follows:

- A set of movement techniques for seated VR, both *Travel* and *Orientation*, which future works may also use as baseline for comparisons;
- A user evaluation that provides information over comfort, confidence and other factors that contribute to end-user experience quality;
- Key points for developing future movement techniques for seated VR (or otherwise), based on the results of our evaluation.

1.5 Publication

An early version of this work was submitted and accepted for presentation in an international peer-reviewed scientific event. The version submitted featured preliminary results only. A version with the full results will be submitted in a future venue.

- Guilherme Amaro and Daniel Mendes. *Travel and Orientation Techniques for Seated VR*. International Conference on Graphics and Interaction (ICGI). 2021. (accepted)

1.6 Document structure

Beyond this point, this dissertation will be divided into six chapters: *State of the Art*, *Proposed Techniques*, *Prototype*, *Evaluation*, *Results*, and *Conclusions*.

In *State of the Art*, we will discuss previous work that relates to our objective. We will look into movement techniques, including movement and input implementations, presence and cybersickness, which are relevant to the VR experience, and ergonomics, which determines its overall comfort. We will also address related work and conclude the chapter with a discussion over its contents.

In *Proposed Techniques*, we describe the selected techniques that we implemented in our prototype. We include visual examples of each technique, measures, and inputs.

In *Prototype*, we show details relevant to the prototype developed for our test. This includes a description of the architecture, hardware setup, the idea behind the developed VE, and VR optimizations used.

In *Evaluation*, we describe the methodology behind the tests, as well as characterize the tasks and its elements. We also defined the participant population.

In *Results*, we show and discuss the results obtained from both objective and subjective evaluations. We made available tables and graphs to improve readability. We also present observations made during those participations.

Finally, in *Conclusions*, we quickly summarize conclusions taken from this dissertation, and discuss future work based of our work.

Chapter 2

State of the Art

Developing and implementing navigation techniques require looking at current existing methods and attempt to comprehend the inherent aspects that affect the user experience. These include the actual travel and orientation of the user, the control scheme, visual feedback, posture, motion sickness, continuity, and many others.

2.1 Current navigation techniques

Navigation techniques usually fall into two categories: Discrete and Continuous. Discrete techniques set the user's orientation or position as specified, without an in-between transition. Continuous techniques rotate and move the user over time.

In addition, each navigation technique can be divided into two parts: Travel and Orientation. Travel allows the user to explore the world, while Orientation aids in the user's orientation in said world.

Finally, each technique implementation is tied to an input method. Depending on several factors, such as ease of use, complexity, and direct feedback, these may change it drastically.

2.1.1 Travel

A frequently proposed method due to its simplicity is Teleportation[9, 5]. It is a simple discrete Travel method between two points and is often mentioned as one of the least prone to cybersickness. Despite this, it is often the least preferred as well, as it lacks the immersion of typical room-experience VR and can cause disorientation due to the non-gradient change in surroundings. Within this method, several variants exist to improve on some of its downfalls, such as the use of transition effects. Additionally, there have been versions that adapt it into a pseudo-continuous

[1] Image taken from Half-Life: Alyx VR info page (https://www.half-life.com/en/alyx/vr)

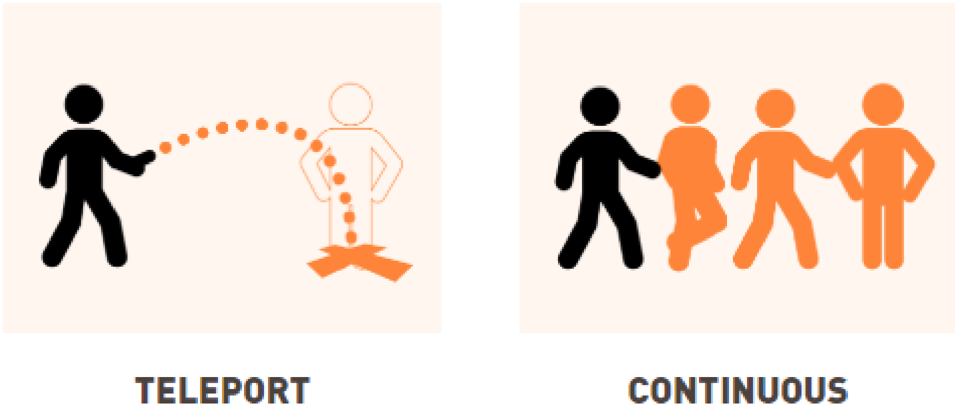

Figure 2.1: Common movement technique examples[1]. *Teleport* is a discrete method of movement, while *Continuous* is as its name implies.

form, such as Pulsed Interpolation[18], by interpolating positions between both origin and destination, or even a fully continuous version, such as Linear Motion[15], similar in nature, but with a smooth movement between points instead.

In contrast with the previous methods, others act on a given direction instead of a specified destination. The user moves continuously as long as the input is held. This method of continuous movement can be used in its raw form, but it is often abstracted to a medium. In one example, the user rides a flying carpet, allowing for both vertical and horizontal travel, fitting a static movement situation[16]. Video games use this type of medium frequently[11], in which context is given for the chosen implementation. These include battle mechs, vehicles, rollercoasters, and others. It appears to reduce disorientation and increase immersion compared to Teleportation. Unfortunately, they often present a significantly higher likelihood of cybersickness.

In addition to the previous methods, functional extensions exist to help in specific parts of movement. One such example is the Jumper metaphor[4]. The typical behavior is natural walking for short travel, and a Teleportation-like mechanic is used to cover large distances. This is achieved by an alternative input method, allowing a choice of either Travel mode on demand. Alternatively, a multi-scale approach can be used for the same effect[1, 6, 14]. Instead of Teleportation, the modeled eye distance is changed to increase the distance perception of their surroundings, making the VE appear smaller and allowing for more substantial translations for the same motion. This method, however, is not always convenient, as indoor navigation requires special workarounds to allow the user to see the environment.

Finally, we can also consider some movement methods not meant for VR during their conception. This is the case of Drag'n'Go[17], which is intended for use on a smartphone. By pointing towards a solid wall or object, the user can then interpolate between that location and their initial position by dragging the pointing finger to the bottom of the screen. While not meant for VR, it can still be used for navigation on a VE.

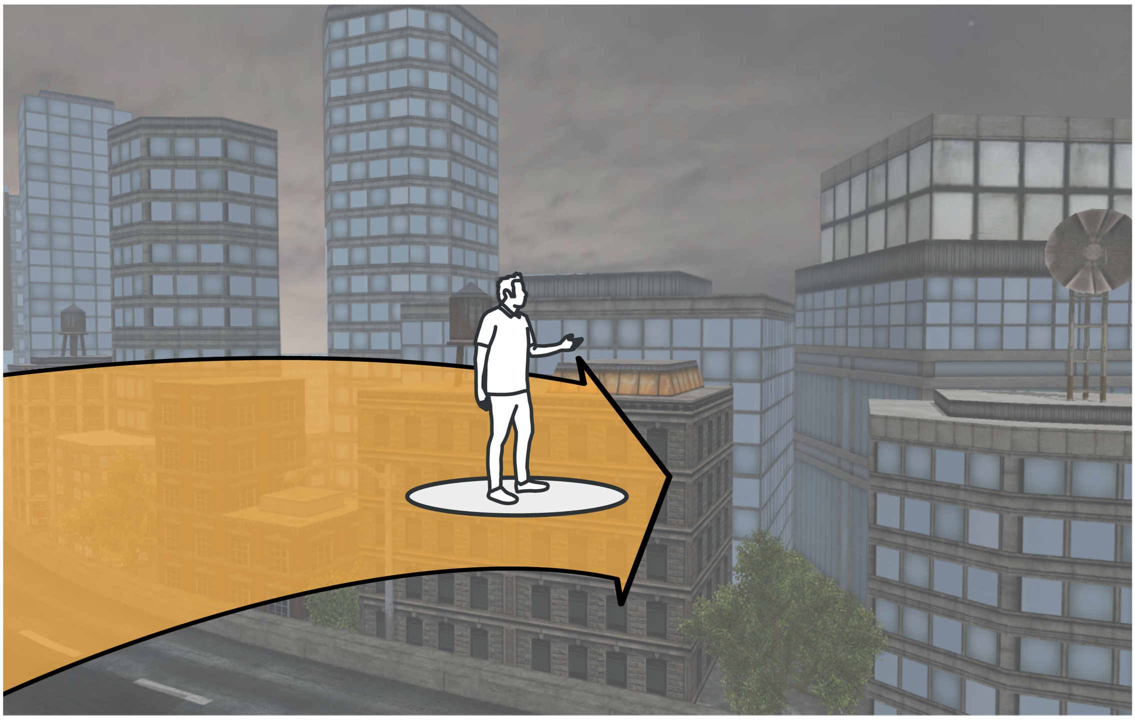

Figure 2.2: Magic Carpet example. From [16].

2.1.1.1 Input implementations

When it comes to input, Discrete travel methods tend to rely on a pointing system or a node-based system. Usually, pointing methods use a VR controller and use a slight arch trajectory to aid in targetting. Then, with a press of a button, the user can apply the translation. Since the pointing mechanism relies on a platform "hit", it limits travel to floor-level sections, as it is impossible to select free space.

For Continuous methods, several more input methods exist, although most were only intended for a horizontal plane of movement. This is the case of the typical "Joystick" movement, a common and straightforward implementation[11]. It can use various hardware solutions and is standard for non-VR applications, such as video games, most notably using analog sticks in console controllers. By pushing an analog towards an intended direction, the user can move towards it at a velocity determined by the distance from the center. Another less known method is "Leaning"[5]. Moving the body towards, away, or to the sides of a reference point results in a similar solution to "Joystick", where the user moves in the respective direction. However, these have very different implications, such as fatigue and physical discomfort, and should not be combined into a single category.

In addition to these methods, gestures can also be used for movement interactions. These can be read through the use of touch surfaces or other sensors that can capture air gestures. For the latter, a Leap Motion or motion capture device is used. One particular implementation — "finger-walking-in-place", or FWIP — distinguishes itself by attempting to map regular walking

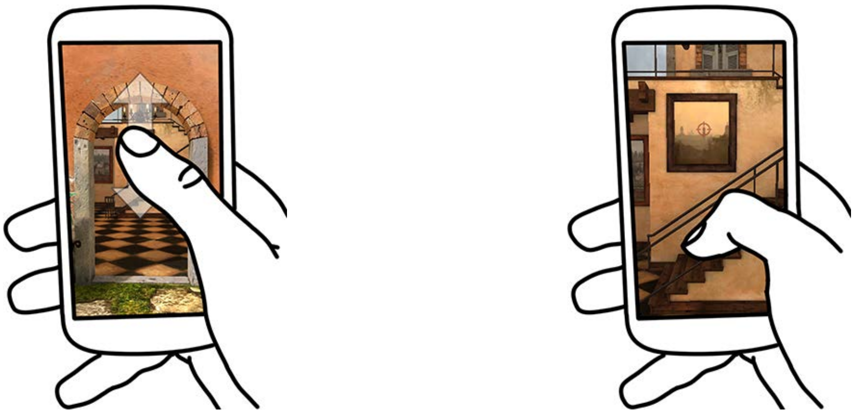

Figure 2.3: *Drag'n'Go* method[17] example

to a motor-equivalent action[12, 13]. Different iterations use different configurations, but the user can use two fingers on a touch surface and simulate walking similar to the movement of legs. Another implementation is the "Airplane" method[3], where the user extends their hands forward and, through Leap Motion capture, moves forward and backward by opening and closing their hands respectively. This implementation also allows for velocity control by finger count in one of the open hands.

There are also input methods based on custom-built hardware. One such example is "Walking by Cycling"[8]. It makes use of cycling biomechanics that translate into walking motion while the user remains seated. This allows for fine control of movement while leaving the hands free for other tasks.

Some of the previous methods require an additional method of specifying the direction of movement. In "Magic Carpet: Interaction Fidelity for Flying in VR"[16] we are introduced to Gaze-Oriented Steering and Hand Steering. They use the head and hand orientation respectively to specify the direction of movement and a button to prompt movement

2.1.2 Orientation

A typical VR experience does not require orientation methods, seeing that natural orientation through the headset is often suficient. However, in certain situations in which the user might be restricted from rotating physicaly, such as when seated, a reorientation method is required to allow access to the virtual world beyond their peripheral field of vision.

Similar to Travel, it also includes both Discrete and Continuous methods. Continuous methods are simple rotations of the user's virtual body, with variations primarily around the input. Discrete methods, however, include specifying the final direction to re-orientate towards, or simply a "stepped" rotation.

2.1.2.1 Input implementations

In "HCI Lessons From PlayStation VR"[11], a few commercial games are explored, with varied reorientation implementations. Some are directly related to the movement medium, such as vehicles. One existing type of implementation is through head input. This is convenient in a VR headset setup, as the hardware required is always present. Taking an initial head orientation into account, looking to the sides will rotate the user towards that side.

Another explored method is the simple rotation through an analog stick, applying to both continuous and "stepped" variants. It can be seen as a "legacy" implementation from non-VR games.

Finally, there were methods of orientation within *Travel* methods, such as *AngleSelect Teleport*, *Curved Teleport*, and *HPCurved Teleport*[9]. On teleportation, the user also selects the final orientation of the virtual body.

2.2 Presence and cybersickness

Affordable VR has brought many different uses for the technology, one of which being entertainment through video games. The amount of information gathered though this medium helps establish patterns to avoid common problems. An article by M. Jacob Habgood et al.[11] analyzes a large pool of games from the PlayStation VR library, including issues associated with the implementations, most notably motion sickness caused by the method of movement and overall user experience. According to the article, "[...]short, fast movements in VR (with no acceleration or deceleration) don't appear to induce significant feelings of motion sickness for most users. This seems to be true of both rotational and positional movement" [p.10, *Conclusion: Short, Fast Movements*]. The same authors further explore this idea in another article[10], including a more specific time span of 300 miliseconds or less.

There have been attempts at using virtual metaphors for real life objects to increase presence and reduce cybersickness and disorientation. Examples include virtual limbs and desks[19], with varying degrees of effectiveness. One such example looked into a variety of factors, including hand representation or realism, extension of the limbs, and their disparities between real world and virtual world counterparts[7].

2.3 Ergonomics

Ergonomics are intrinsic but not exclusive to VR. A lot of information can be obtained by exploring adjacent works. For example, M. Bachynskyi et al.[2] has investigated performance and ergonomics of common surfaces, such as smartphones, tablets, laptops and large displays. The results can still be used for reference when working on a VE.

However, work directly related to VR has also surfaced. This includes study of matching virtual and real metaphors, and VR menus aligned with real surfaces.

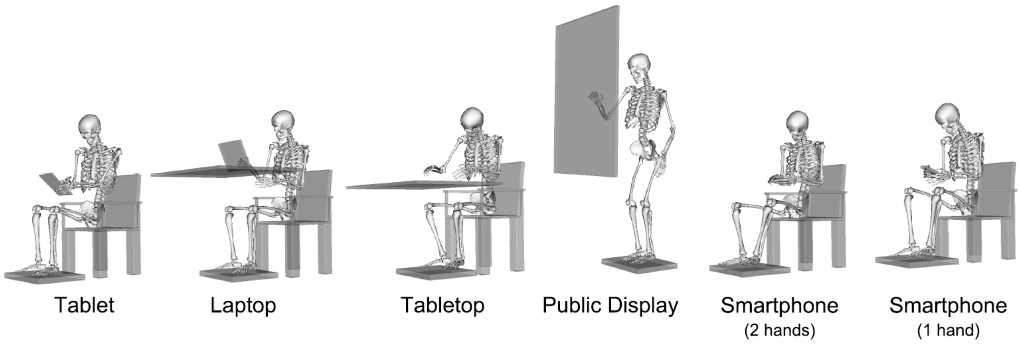

Figure 2.4: Example of postures during interaction with surfaces[2]

2.4 Discussion

In order to analyze each implementation, we evaluated each of the following categories:

- *Custom Hardware*, where the implementation requires specific hardware for that use only;

- *Free Movement Support*, which indicates if the user is restrained to pre-specified locations of the scene;

- *Sideways Movement Support*, where we state if the implementation allows for independent *Travel* perpendicular to the user's direction;

- *Vertical Movement Support*, similar to the previous, but for vertical movement;

- *Separate Movement Support*, where the *Orientation* implementation is not tied to the *Travel* implementation of the user;

- *Continuous Movement Support*, where *Travel* is not discrete, but continuous.

- *Seated Support*, where the implementation can be used in a seated experience.

With the evaluation complete, we saw that certain methods would not be feasible for the context of this thesis. This is the case of some of the *Magic Carpet* implementations that require a non-seated experience. The same could also be said for *Walk by Cycling* due to the need for specific hardware, and for *Node-Based* due to its restrictive nature. Additionally, although not specified in the evaluation, Buttussi et al.[5] looked into the *Leaning* method, and found no significant improvements over *Teleport* and *Joystick* while bringing other aspects into account such as spine fatigue. As such, this method may not be worth considering given time and complexity constraints.

Point & Click techniques seem to be very similar to each other. They offer fairly good positional control, and are not prone to motion sickness as much as other implementations. They serve as a good benchmark for comparison. However, in this project, we focused on continuous methods of *Travel*, and as such, *Teleport* was not used. With this in mind, however, some specific variations were taken into account, such as the *Orientation* mechanic demonstrated by Funk et al.[9], as they show good acceptance by participants.

Table 2.1: Movement methods

Method	Input Type	Custom Hardware	Support					
			Movement					Seated
			Free	Sideways	Vertical	Separate	Continuous	
Teleport[15]	Point & Click		•	•	•	•		•
Linear Motion[15]	Point & Click		•	•	•	•	•	•
Animated Box[15]	Point & Click		•	•	•	•		•
Pulsed Interpolation[18]	Point & Click		•	•	•	•		•
Teleport w/Orientation[9]	Point & Click (w/ orientation)		•	•	•			•
Node-Based[10]	Point & Click (towards node)							•
Joystick[5]	Directional Input		•	•		•	•	•
Leaning[5]	Leaning		•	•			•	•
Human Joystick[16]	Speed circle		•		•		•	
Joystick (Magic Carpet)[16]	Directional Input		•		•		•	•
Human Joystick[16]	Walk-in-Place		•		•		•	
FWIP[13]	One-handed Gestures		•				•	•
FWIP[12]	Two-handed Gestures		•			•	•	•
Walk by Cycling[8]	Cycle-Based	•	•			•	•	•

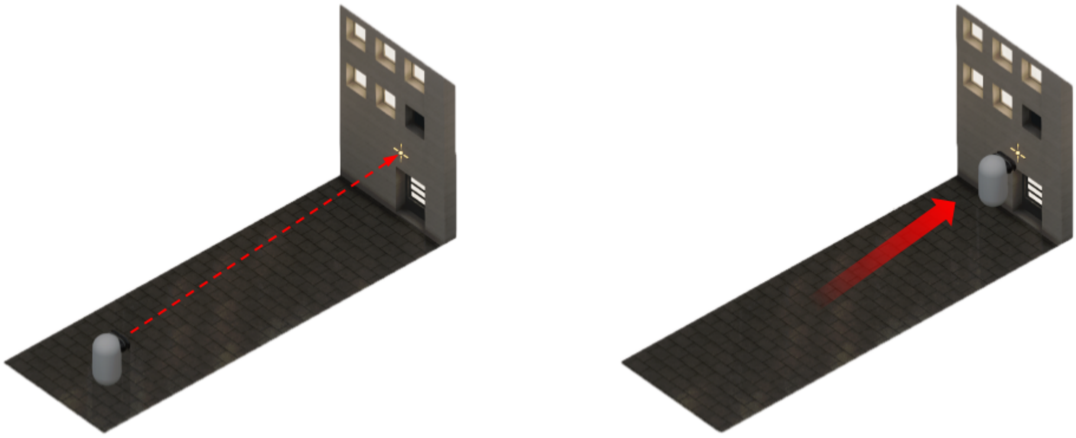

Figure 2.5: *Drag'n'Go* example in VR

Directional Input techniques, often known as *Joystick* techniques are very present in video games[11], and also appear in one of the *Magic Carpet* implementations[16]. They usually suffer from significant motion sickness, and acceptance seems to vary by implementation. This could be due to the fact that many *Joystick* implementations also use *continuous rotation*.

One last method worth considering is *Finger-Walking-in-Place*, especially with two-handed input. It should allow more precise control compared to the previous methods. We prefer the two-handed input, as we focus more on navigation rather than interaction. That said, early exploration of the technique showed possible ergonomic issues related to hand and wrist position, so alternatives were explored.

VR methods aside, the *Drag'n'Go* method proposes an interesting alternative to the other techniques, and can be adapted into a VR environment through use of the user's gaze.

Chapter 3

Proposed techniques

For this project, we propose a set of VR locomotion and orientation techniques for when the user is seated. Some are adaptations of currently existing techniques which were not originally designed for this scenario, while others are new proposals.

3.1 Locomotion

For locomotion, we present three techniques, all Continuous, as they seem to be most adequate for exploring the VE while maintaining spatial awareness and collecting information about the user's surroundings.

3.1.1 Continuous Directional Movement

The Continuous Directional Movement technique (Figure 3.1) consists on the user pointing a VR controller in the intended direction of movement. The method uses the controller's direction as the

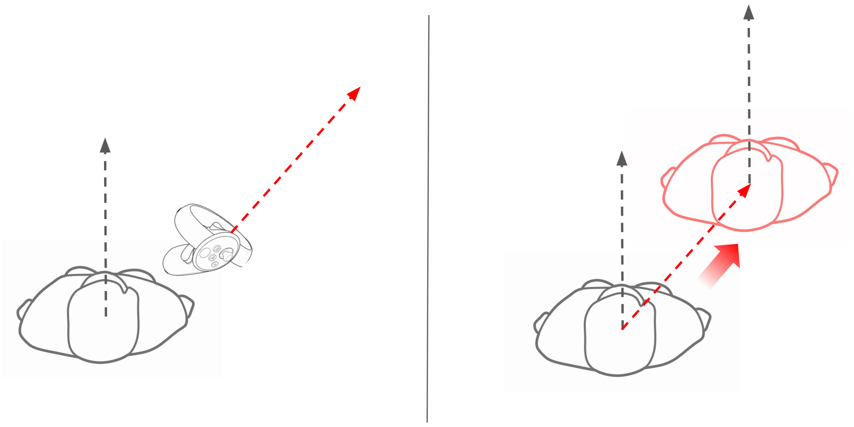

Figure 3.1: Continuous Directional Movement locomotion technique

movement's direction and, when pressing the trigger, the user will move in the indicated direction. The intensity of the trigger pull determines the final velocity. When fully pressed, the user moves at 5 m/s in the intended direction, decreasing linearly to a full stop with the full release of the trigger.

3.1.2 *Dog Paddle*

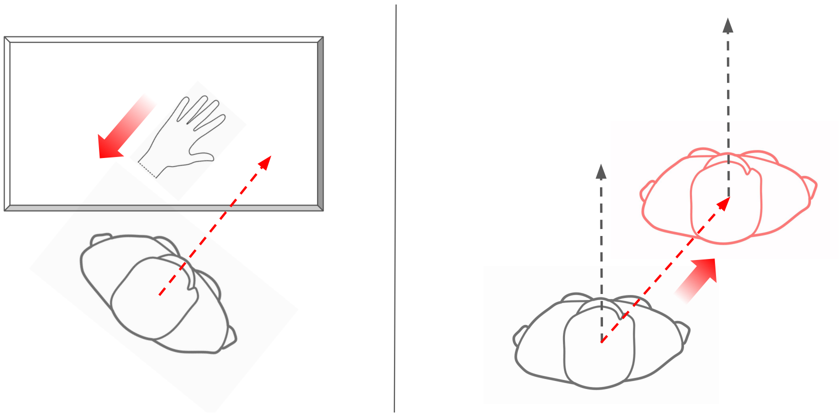

Figure 3.2: *Dog Paddle* locomotion technique

The Dog Paddle technique (Figure 3.2) consists in the use of simple gestures over a touch surface to move the user in the direction of of their gaze. In practice, it's based on the same concept of motor equivalence behind the FWIP technique, but with a higher focus on its ergonomic aspect. The touch surface reads the movement of each dragging hand (identified by five touch points near each other), similar to a person's swimming gesture or "paddle", and directly moves the user proportionally, which is pre-defined. Each 40 cm of hand travel (which is the height of the interactive touch surface used in this test) corresponds to a movement of 65 m in the VE. Smaller or larger movements are mapped linearly. Head direction is taken into account when input is received from the surface to account for possible misalignment with hand movement, as the user cannot see their hand gestures directly.

3.1.3 Drag'n'Go

This method is an VR adaptation of the Drag'n'Go approach. Similarly to the previous technique, this method reads the user gesture to move in the user's gaze direction (Figure 3.3). However, the final position is determined the moment the gesture begins, resulting from a ray cast from the user's head, inheriting its direction, to the first point of collision with any object of the scene. The end point will be fixed until the hand gesture ends, that is, until the user stops touching the touch surface. Until then, the user can drag the hand around, to the bottom of the surface to reach the end point, or above the initial contact point to pull away from the starting position. The distance between the starting touch position and the bottom end of the interactive surface is mapped to the distance between the original VE position and the ray cast target respectively. This way, the higher

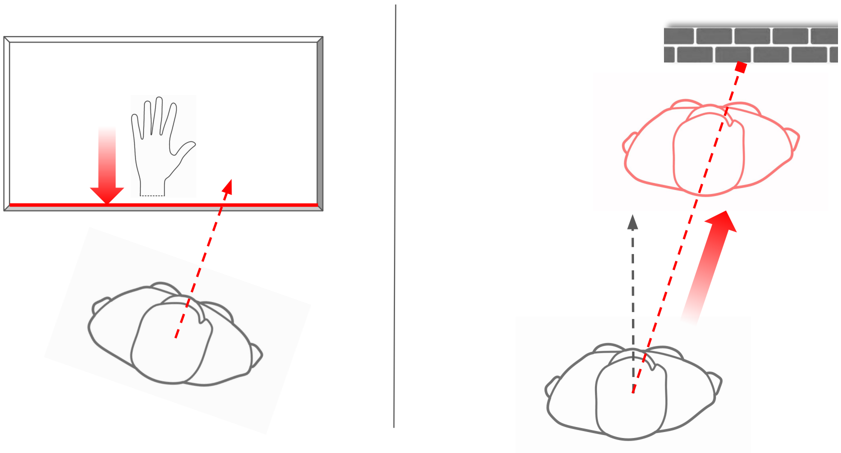

Figure 3.3: *Drag'n Go* locomotion technique

the gesture starts on the surface, the greater the degree of precision and the slower the movement becomes. Should the gesture start while not looking at a valid object or wall, the end point is defined as 50 meters forward from the user's perspective.

3.2 Orientation

As far as orientation is concerned, we present four techniques, two continuous and two discrete.

3.2.1 Continuous Directional Rotation

The Continuous Directional Rotation technique (Figure 3.4) consists in the user indicating the direction of rotation, which is applied over the vertical axis of the user, through the use of an analog stick present on the VR controller. It is possible to indicate the direction and intensity. By pushing the analog stick completely to the left, the user rotates to the left at a rate of 60 °/s, and vice versa by pushing to the right. The rotational velocity changes linearly through the horizontal axis of the stick, and the vertical axis is ignored.

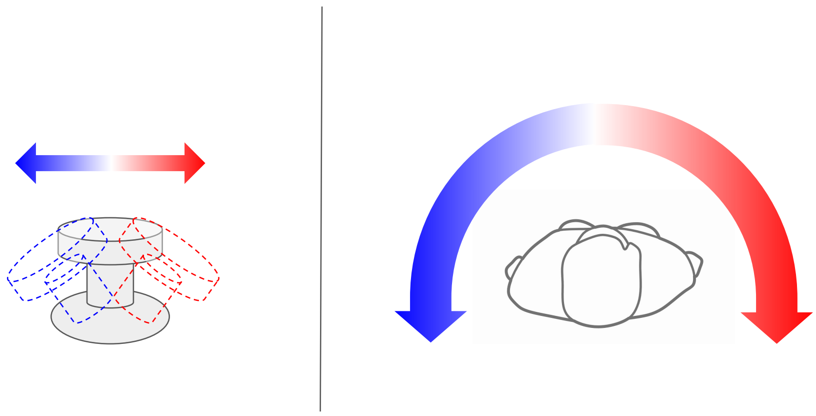

Figure 3.4: Continuous Directional Rotation technique

3.2.2 Tactile Surface Dragging

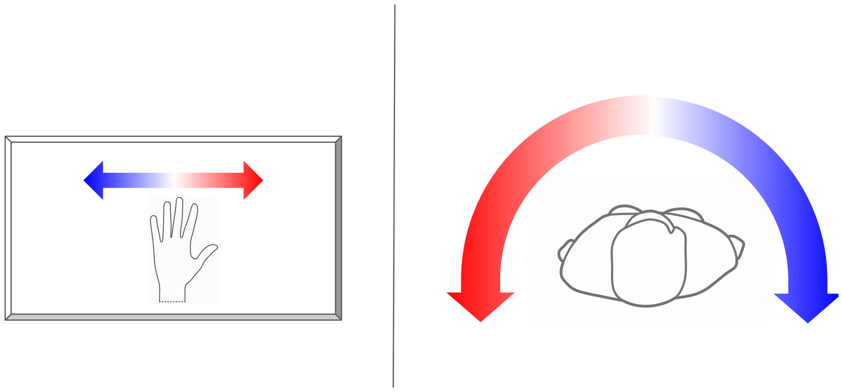

Figure 3.5: Tactile Surface Dragging technique

The Tactile Surface Dragging technique (Figure 3.5) consists in reading a gesture over a tactile surface and translate that gesture into a rotation over the vertical axis of the virtual body. By dragging one hand horizontally over the surface, the user proportionally rotates towards the opposite direction, similar to the regularly used smartphone swipe. For a full horizontal swipe, the user rotates 160º over the 71cm of total width of our test tactile surface.

3.2.3 *Choose & Click*

The *Choose & Click* technique (Figure 3.6) consists in pointing towards the intended final direction of rotation through the use of the analog stick present in the VR controller. The final direction is defined by the vector formed between the center of the analog and the final position of the stick. After defining the directions, and without letting go of the stick, the user can then press a button to confirm the rotation. This rotation is done in 150ms, according with a previous article, regardless of its amplitude.

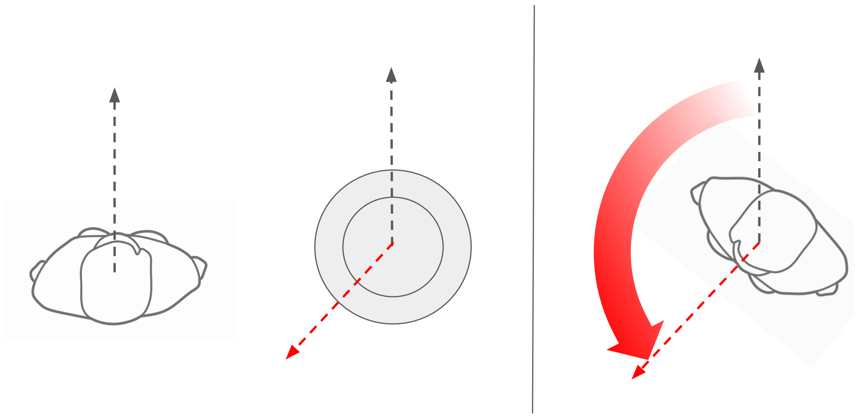

Figure 3.6: *Choose & Click* technique

3.2.4 Gaze Convergence

Similarly to the previous technique, the Gaze Convergence technique (Figure 3.7) uses the final orientation of the head instead of a analog stick. The final direction of the virtual body is defined by the current direction of the head, allowing the user to simply look in the intended orientation. This way, by confirming the action with a button, the user's virtual body rotates towards to original orientation of the head. Like the previous technique, this rotation is executed within 150ms.

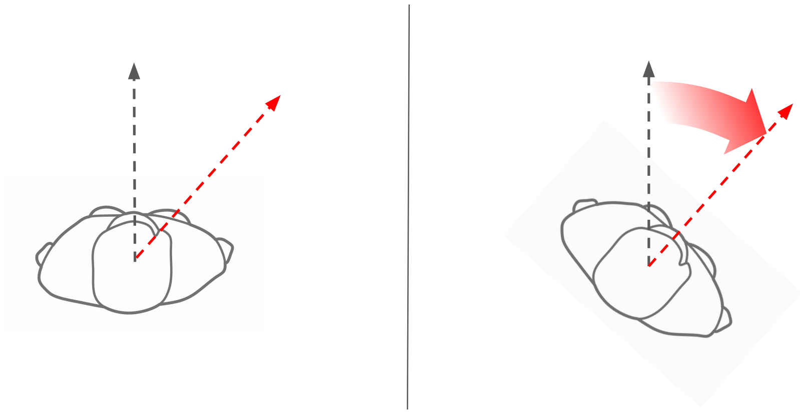

Figure 3.7: Gaze Convergence technique

3.3 Summary

In this chapter, we proposed 3 *Locomotion* and 4 *Orientation* techniques to be evaluated. For *Locomotion*, we have *Continuous Directional Movement*, which makes use of controller pointing to continuously move in the desired direction, *Drag'n'Go*, which moves the user towards the target of their gaze, and *Dog Paddle*, which propels the user with each "paddle" gesture they make on the touch surface. As for *Orientation*, we proposed *Continuous Directional Rotation*, a common and simple joystick implementation for rotation, *Tactile Surface dragging*, which uses the horizontal gesture movement to rotate the user accordingly, *Choose & Click*, which allows the user to select a final orientation based on joystick input, and *Gaze Convergence*, which is similar to the previous, but uses the gaze to determine final orientation.

Chapter 4

Prototype

In order to compare the multiple proposed techniques, we developed a VR application where participants can move around with specific evaluation tasks.

4.1 Hardware Setup

The hardware setup we used consisted on a Oculus Rift S headset with hand controllers and a multi-touch surface frame. The headset is capable of a 80hz refresh rate, with a field-of-view of 115º, and uses inside-out tracking. The touch surface uses infrared sensors to track each finger as it crosses the beams.

4.2 Architecture

As previously planned, we opted to use Unity to develop the VR application. Although alternatives such as CryEngine, Unreal Engine, and Godot exist, Unity appeared to provide the best hardware support, performance, and ease-of-use in our case. Additionally, it was a platform we had experience with, further reducing development time.

Within Unity, we create Scenes, each defining a VE. Each scene has objects, which Components characterize. One of those objects is the Player. It has a Camera (or a child object with the Camera component in this case), which renders an image for the HMD, and a PlayerController class, which defines the user's behavior inside the VE. In order to read the user's input, PlayerController uses an InputLayer instance, which reads all input sources and manages them accordingly, returning the resulting movement intention and switching the visual interfaces as needed. InputLayer makes use of Unity's new Input System to read those inputs, from controller buttons and movement to touch input.

Another type of object is the CourseController. These have checkpoint objects called CourseObjective as children and follow the rules set by CourseRuleSet, which state what units are to be measured, what inputs should be read, and what behavior to adopt.

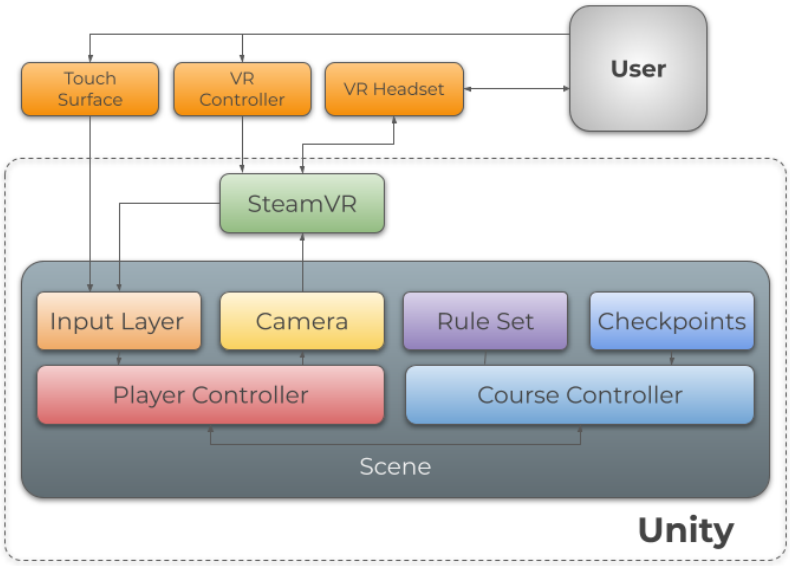

Figure 4.1: Architecture diagram

4.3 Virtual Testing Environment

We built our Virtual Testing Environment with many different requirements in mind:

- **Scalability**: the environment should simulate an actual use case of the techniques. We opted to use a Unity Store Asset for this end, hoping that it would save development time (which it might not have, see "VR Optimization"). The asset in question was a low-poly cityscape with streets, alleys, and backyards.

- **Performance**: establishing a connection between a poor, jerky experience and cybersickness is not part of this project's objectives, and as such, we want to avoid unnecessary variables. The VE should be performant enough so that the testing hardware should always maintain the maximum framerate for the given VR headset. This is better detailed in the "VR Optimization" section.

- **Visibility**: the task objectives and interactable objects should be easily visible and striking. This required some changes to the overall scene ambiance to contrast the objectives and the scenery.

Figure 4.2: Final version of the Virtual Testing Environment

- **Experience quality**: it was expected from the start that a large portion of our participants would be using VR for the first time, and as such, it could pose as an opinion-defining first experience. Details such as building variety, illumination, high-quality textures and effects, reflections, and animations were added to make the experience pleasing and entertaining.

4.4 VR Optimization

One of the biggest hurdles of this project was adding features while reducing the risk of performance drops, which was accentuated by the quality requirements of the VE. As such, a lot of development time went into optimization.

By default, Unity includes a lot of handy optimizations for VR and otherwise. One of the simplest is rendering the VR image in "Single Pass Instanced". This allows both eyes to be rendered in a single pass, reducing draw calls and time taken culling geometry. It has some limitations regarding platform support, but since the testing hardware is controlled and compatible, this was not an issue. Another optimization method is the use of "light baking". Instead of calculating and rendering shadows in real-time, possibly from hundreds of different sources, lights and ambient occlusion are "baked" into textures, which are applied on top of static objects, substantially reducing frame render time and even adding light bounce for better quality illumination. Since we do not use moving objects other than the Player, this was a straightforward performance improvement. However, "baking" is time-consuming, and depending on the hardware available, could take upwards of days to complete. The rendering process was also somewhat touch-and-go. There were often visual glitches, either due to texture resolution or wrong 3D model parameters.

In order to further reduce draw calls, Unity makes use of "batching". This can be done in one of two ways: static batching, where it combines static objects using the same material into a large,

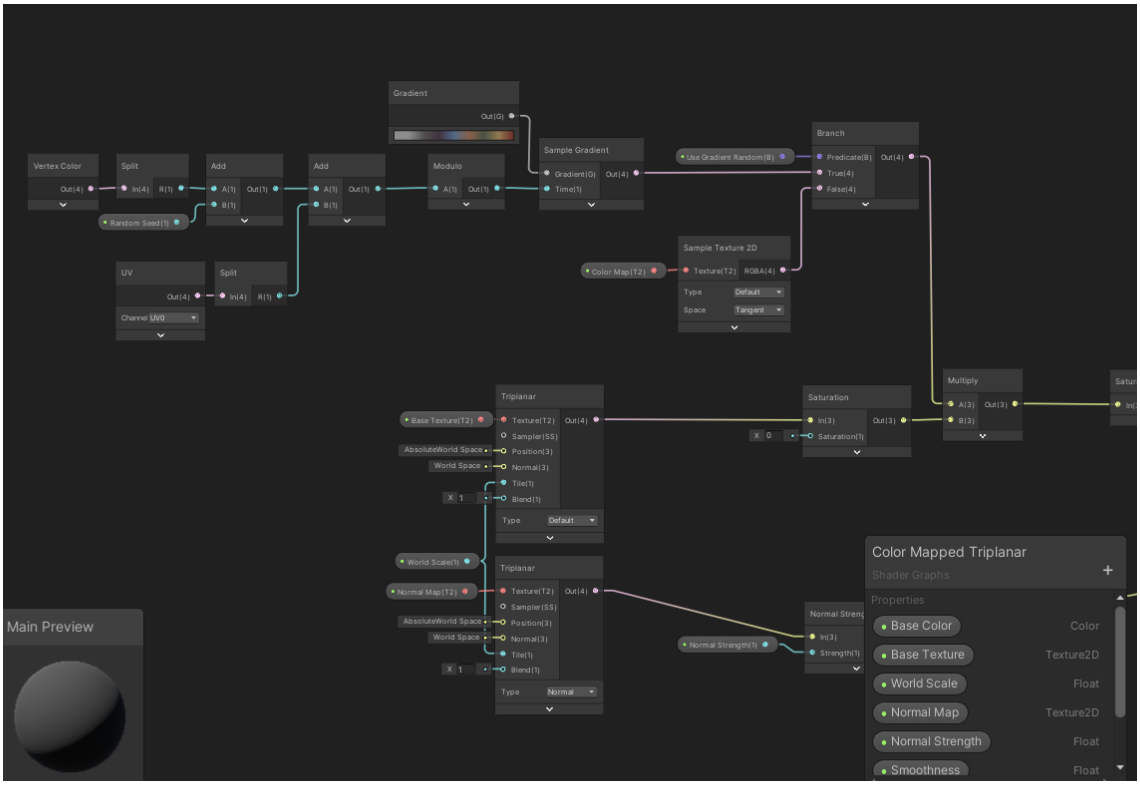

Figure 4.3: Example of the shader used for buildings

single mesh, or dynamic batching, which also batches multiple objects with the same material into a single draw call, but certain restrictions apply. Additionally, these materials also have an option to enable GPU Instancing, where it tries to render all copies of the same mesh from the same batch at once, greatly reducing render time. These optimizations took a toll on development time, however. Most objects now needed to use a single material for all instances, so "mega-shaders" were created to use assigned vertex colors for distinguishing buildings. Static batching caused some issues with custom shaders, including incorrect positional values, not batching correctly, and required workarounds to work correctly, including selecting objects to be batched manually through code.

Finally, there is also Occlusion Culling, both using Unity's implementation and a smarter, own alternative. Unity's implementation allows a static version of occlusion, where we can define static occluders that will skip the rendering of smaller objects behind them. It also hides objects outside the camera's view frustum. However, since each building was a single entire mesh, this meant many unseen surfaces were still being rendered, coupled with the fact that the objects were poorly optimized. To fix this, we rebuilt all 14 building variants, tremendously reducing polygon and vertex count. Each side of the building was also cut into a separate mesh. Then, we created a script to disable the renderer for a given mesh whenever its surface was not visible from the camera's current angle. Some other objects were instead merged into a single mesh of the same type, such as roads, pavements, and windows, as they were either too large to be effectively culled or too many. In those cases, occlusion would often induce a performance penalty, and simply

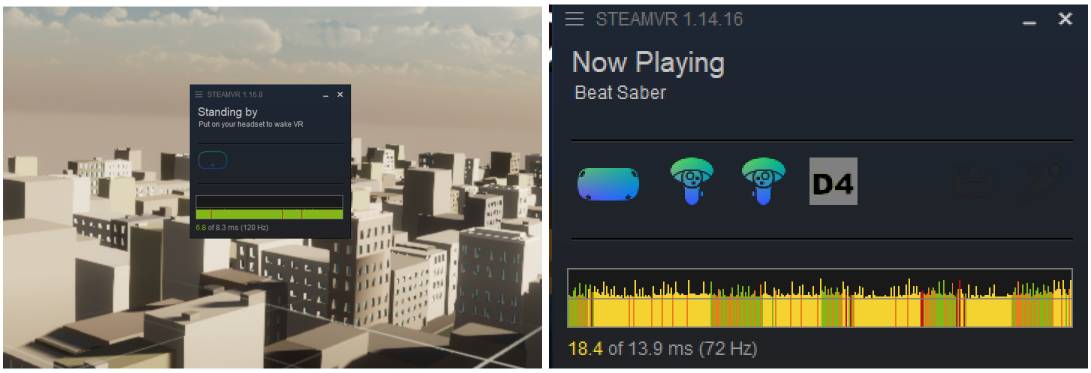

Figure 4.4: Examples of performance graph available through SteamVR. Left image shows an early version of the prototype with stable performance, albeit with a few dropped frames; Right image shows an example of poor performance[1]. Green means no issues, yellow means a previous frame was re-projected for taking too long to render, orange means the frame was rendered late without re-projection, and red means the frame was skipped entirely, resulting in very noticeable stutter.

reducing the draw calls yielded a better result.

Another point of note, although not strictly an optimization, was the particle system used for checkpoints. The translation checkpoints consist of a cylinder with particles spinning on both ends. This amount of particles would not be possible using Unity's default particle system, which leverages the CPU. Instead, thanks to the new tools available with HDRP, we used Unity's VFX Graph, which made use of the GPU and allowed a much higher magnitude of simulated particles.

4.5 Development Notes

During prototype development, there were a few features of note that should be mentioned, some of which were cut from the application altogether.

During the initial phases of development, we implemented a see-through floor system that allowed the user to retain a sense of speed during flight. This grid would accompany the user whenever he was floating above the ground, and would shift accordingly with movement. The idea behind this was that by giving the user a point of reference for "visualizing" speed while flying, it could reduce some degree of motion sickness. We also had implemented a *vignette* mask that would fade in and out depending on user speed and rotation to combat cybersickness. These were scrapped, however. These measures were meant to combat issues, but from what we experienced, it would reduce immersion and overall experience quality. At the same time, any techniques that would not suffer from the combated issues would be penalized, as they would not gain anything from these features, while still suffering penalties from implementation. In the end, we opted to use the techniques as intended, as it would result in a fairer comparison.

[1]Image taken from Reddit post (https://www.reddit.com/r/OculusQuest/comments/jljaa2/oculus_link_lag_on_steamvr_q2/)

Also worth mentioning is the interpolation system added to touch surface tracking. During testing, it became apparent that the touch frame hardware had "skipping" issues, where it would skip between a certain distance or angle. This was visually jarring, and could become a source of discomfort. As such, touch input is interpolated over 3 frames, or 37.5ms at most. This greatly smoothed the visual feedback of input, while remaining unnoticeable.

4.6 Summary

In this chapter, we described the prototype developed for our evaluation, including choice of hardware and software. We used Oculus Rift S and a multi-touch surface frame for the hardware, and Unity and SteamVR for the software.

Within Unity, we created a Scene, which is our VE, and defined a *PlayerController*, which handled information obtained through an *InputLayer* class. The tasks were handled by a *CourseController* class.

We also described our VE requirements, those being *Scalability*, *Performance*, *Visibility*, and *Experience Quality*. With this, we created a city environment with a darker ambience, and some appealing effects, while retaining the maximum performance achievable through the headset.

Finally, we mention some optimizations used to make this possible, such as *Light Baking*, *Batching*, and *Occlusion Culling*, as well as development notes regarding scrapped features, and technical aspects like touch input interpolation.

Chapter 5

Evaluation

In order to establish a comparison between methods, we need to evaluate them. With the list of techniques and prototype set, we have defined the evaluation environment, methodology and variables to look into. In this evaluation, we once again split into two categories: *Travel* and *Orientation*. *Travel* is dependent on *Orientation*, and testing every combination would prove very taxing, both in development and testing duration. Therefore, we chose to split them.

5.1 Methodology

Due to the many different conjugations of Orientation and Travel methods, the experiment is split into two categories. Each category will have its own tasks, done separately and repeated for each method. After the last task of each technique, the participant fills a questionnaire. The participant was allowed to test the method freely for a short amount of time before beginning the set of tasks, usually less than a minute. This allowed them to familiarize with the technique at hand.

For each task, we partially implemented standardized questionnaires such as *SUS*, *NASA-TLX* and *SSQ*. However, since there were quite a few methods to test, and each questionnaire is fairly long (the final version has 16 questions per technique, 123 questions total, with varying degrees of complexity), the amount of questions was reduced to the essentials, as our focus was aimed towards comparison between each method and not making an empirical evaluation of each individual technique. In addition, we made observations on user behavior, frequency of issues, and input exploration, as these might indicate potential issues with our implementation of a certain method, rather than an issue with the movement technique itself. We also recorded task performance, measuring time, distance traveled, and rotation covered. These were obtained directly through the developed prototype, starting at the beginning of each task, when the user commences movement.

The order of the techniques tested by each user followed a Balanced Latin Square distribution to make sure it did not affect test results. As such, we required a minimum of 12 participants.

However, in every case, the user begins with the tasks for all *Travel* techniques, then *Orientation* tasks, and at the end, they test the junction of their favorites from each category. This is done for observation purposes, no tasks are completed at this time.

5.2 Tasks

For our evaluation, we created a total of 4 *Travel* and 3 *Orientation* tasks. These repeat for each technique. During each task, the user is given the current checkpoint direction and order. Depending on the task category, either *Travel* or *Orientation* is locked, always opposite to the one being tested. When *Orientation* is locked, the user can still look around, but the virtual body will not rotate. When beginning a task, the user is teleported to the starting point and both categories of movement are locked. When they give intent of movement, as in, they input an action that would make them move in a specific task, the respective movement is unlocked, and they can perform the task. Time is counted from this moment until the completion of the last checkpoint.

5.2.1 Travel

Figure 5.1: *Travel* task example. The heads-up display shows the current checkpoint number, with the arrow on top pointing towards it.

The tasks used to test *Travel* methods feature a set of cylindrical pre-ordered checkpoints, which the user must go through to complete the task. In order to complete a checkpoint, the participant must come to a full stop inside it. On success, the user receives an audio cue, the current checkpoint fades, and the next checkpoint changes color to yellow. Any remaining checkpoint not yet active appear in light blue. At no point during the task is the participant allowed to change the virtual body orientation, as these tasks serve to test *Travel* exclusively. Each checkpoint should be visible within the periphery of vision, with few exceptions, thus not requiring the user to look around. They should be positioned in front of the user, since it is not possible to re-orientate.

The first *Travel* task is a mixed set of vertically- and horizontally-spread checkpoints. It avoids focusing on specifics, testing natural movement and allowing further familiarization with the technique. The theoretical total distance the user must cover should be around 554 meters, with a span of 140 meters horizontally and 139 meters vertically.

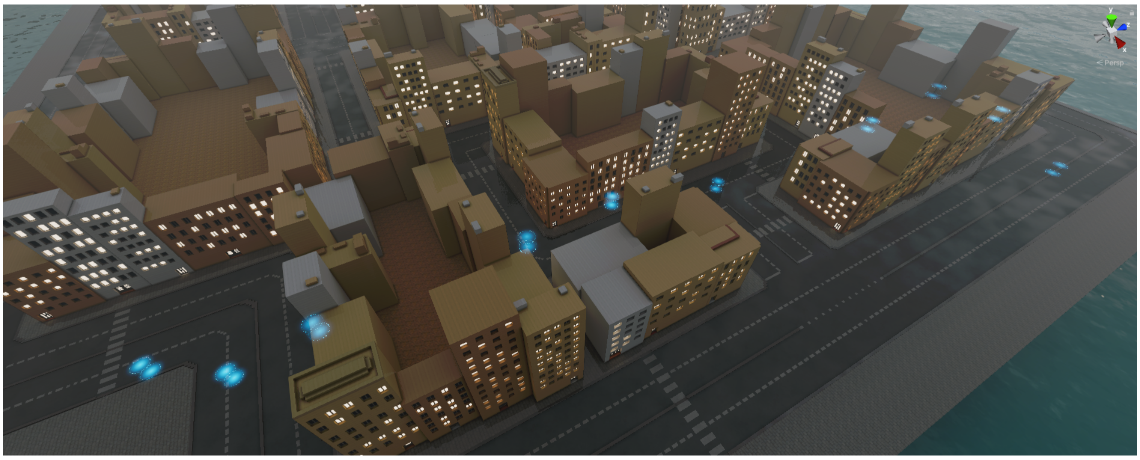

Figure 5.2: Mixed *Travel* task, starting on the right-most checkpoint

The second *Travel* task is mostly a set of horizontally-spread checkpoints. It focuses on normal planar movement. The theoretical total distance the user must cover should be around 433 meters, with a span of 150 meters horizontally and 0 meters vertically.

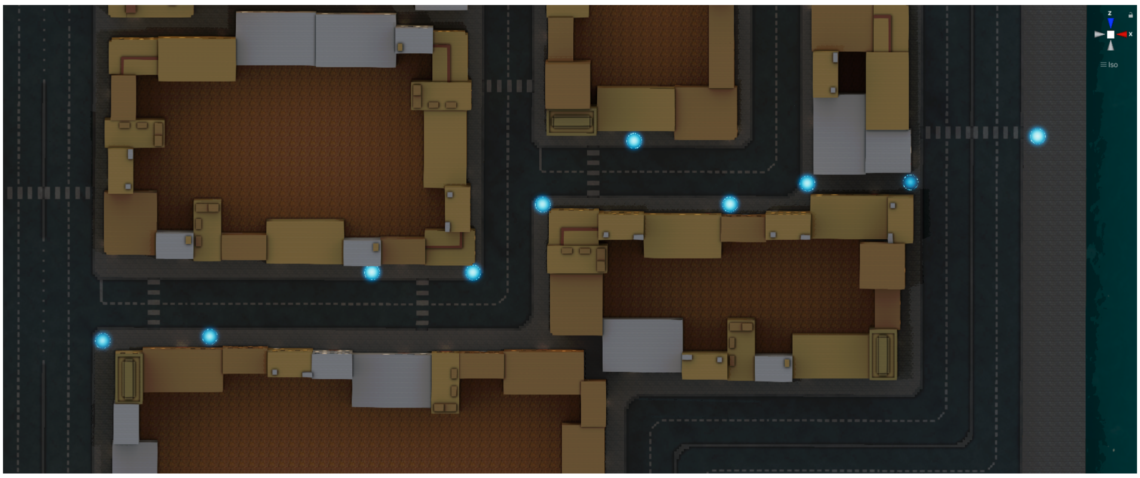

Figure 5.3: Horizontal *Travel* task, starting on the right-most checkpoint

The third *Travel* task includes a set of vertically-spread checkpoints. It focuses on height adjustment and bypassing vertical objects. The theoretical total distance the user must cover should be around 297 meters, with a span of 21 meters horizontally and 113 meters vertically.

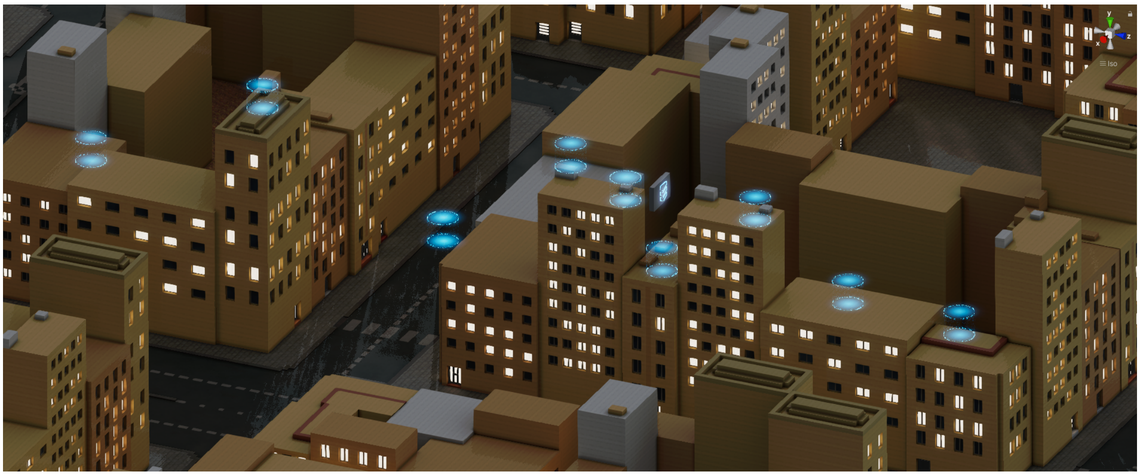

Figure 5.4: Vertical *Travel* task, starting on the right-most checkpoint

Finally, the fourth *Travel* task, also known as the free long task, uses only a single checkpoint on the opposite side of the city from the start, 950.1 meters apart from the start. The user is only told to reach it as fast as possible by any means. This task is focused on technique learning capabilities and input exploration, as well as overall experience, and prolonged flight movement.

5.2.2 Orientation

The tasks used to test *Orientation* methods feature a set of traffic-sign-shaped checkpoints which the user must align their virtual body with. Unlike the *Travel* tasks, these checkpoints are randomized in order. They still retain the same rotation from previous orientation and total rotation, however, theoretically resulting in similar completion times. In order for the user to complete a checkpoint, they must align their virtual body with the pole of the sign within an error of 2 °. To help with this, a visible line extends from the virtual body, reaching any checkpoint the body may be aiming towards. When this line matches with the pole of the current checkpoint, the pole will fill with a yellow line, and outline the sign. This action takes 2 seconds to complete, and can be interrupted if the user misaligns the virtual body with the checkpoint, setting the checkpoint to its original state.

The first *Orientation* task is a mixed set of both small and large rotations, resulting in a total of 540 ° of rotation over 7 checkpoints. This task does not have a particular focus when it comes to evaluation. It should act as a baseline for the other tasks.

The second *Orientation* task is a set of small rotations only, for a total of 175 ° of rotation over 7 checkpoints. This task focuses on evaluating short and precise rotations.

The third and last *Orientation* task is a set of large rotations only, for a total of 490 ° of rotation over 4 checkpoints. This task focuses on evaluating quick changes of direction.

Figure 5.5: *Orientation* checkpoint (right), *Orientation* task example (left)

5.3 Participants

We gathered 12 participants, of which 7 were male and the remainder female. 8 were aged between 18 to 25, 3 were between 26- and 32-years-old, and 1 was in the 49 to 62 category. 8 did not have previous experience with VR, 2 experienced it very infrequently (less than once per month), and 2 used VR daily. Only 2 participants mentioned known issues with vertigo. 2 users did not experience any symptoms of cybersickness during testing, both of which where VR first-timers.

Figure 5.6: Participants using VR controller (left) and touch surface (right)

5.4 Summary

In this chapter, we describe the method of evaluation we used. We split the evaluation into two categories, *Travel* and *Orientation*, repeating a set of tasks and a questionnaire for each evaluated method. We also recorded time and distance traveled for each task, and made observations on user behavior, input exploration and issues. We also describe each task individually. We start with *Travel* tasks, which are a set of checkpoints to be traversed in as little time as possible, and do not allow for *Orientation* methods. These include a mixed vertical and horizontal course, a mostly horizontal course, a mostly vertical course, and a long distance free flight course. As for *Orientation* tasks, it features a set of same-total-rotation checkpoints where the user is required to align the virtual body with the checkpoint. These courses feature a mixture of small and large rotations, then small rotations only, then large rotations only. Finally, we also give a brief description of our participants. Most were aged between 18 to 25 and did not have any previous experience with VR.

Chapter 6

Results

Throughout this chapter, we will show the results from the data collected during the evaluation phase. This includes objective user performance, as recorded through the prototype, questionnaire results and observations made during tasks.

6.1 User Performance

As previously mentioned, we collected task performance data through our prototype. More specifically, time elapsed for both categories of tasks, total path length for *Travel* tasks only, and total rotation for *Orientation* tasks only. This data is continuous, so we attempted a one-way ANOVA with repeated measures where possible. To determine normality, we conducted a Shapiro-Wilk test. We then determined sphericity through Mauchly's Test of Sphericity. In the cases the data was not spherical, we applied Greenhouse-Geisser correction. If it was determined that the data was not normally distributed, we instead opted for a Friedman Test, and Wilcoxon Signed-Rank Test post-hoc when possible. All post-hoc results were applied a Bonferroni adjustment within their own category (3 possible pairs for 3 *Travel* methods, 6 possible pairs for 4 *Orientation* methods).

Task	Time Mean (s)			*Friedman Test or ANOVA*
	CDM	Drag'n'Go	Dog Paddle	
Mixed	37.062 ^{a}	60.904 $^{a\ b}$	38.588 ^{b}	$F(2,16) = 12.523, p < .001$
Horizontal	24.312 ^{a}	37.537 ^{a}	31.857	$\chi^2(2) = 11.167, p = .004$
Vertical	23.383 $^{a\ b}$	48.416 ^{a}	41.386 ^{b}	$F(2,18) = 30.062, p < .001$
Free Long	39.642 $^{b\ c}$	17.399 $^{a\ c}$	23.366 $^{a\ b}$	$F(1.192, 10.729) = 72.369, p < .001$

Table 6.1: *Travel* technique times, $^{a\ b\ c}$ mean pair of statistic significance.

Starting with *Travel* times (table 6.1), in the mixed task, *Drag ´n ´Go* was statistically significantly slower than both *Continuous Directional Movement* ($p = 0.12$) and *Dog Paddle* ($p = .020$).

However, in the horizontal task, only *CDM* was significantly faster than *DnG* ($Z = -3.059, p =$ .006). In the vertical task, *CDM* was much faster than both *DnG* ($p < .001$) and *DP* ($p < .001$). Finally, on the long task, *DnG* was significantly faster than both *DP* ($p = .044$) and *CDM* ($p < .001$). *DP* was also faster than *CDM* ($p < .001$) this time.

Task	Measurement Mean (m)			Friedman Test or ANOVA
	CDM	Drag'n'Go	Dog Paddle	
Mixed	582.537 [a]	852.224 [a b]	600.532 [b]	$F(1.088, 6.526) = 19.491, p = .003$
Horizontal	434.452 [a]	646.885 [a b]	448.058 [b]	$F(1.019, 6.113) = 16.332, p = .006$
Vertical	330.041 [a c]	489.938 [a b]	394.681 [b c]	$F(2, 14) = 23.463, p < .001$
Free Long	984.950	1058.580	1031.268	$F(2, 18) = 3.896, p = .039$

Table 6.2: *Travel* technique measurements, [a b c] mean pair of statistic significance.

As for path length (table 6.2), in the mixed task, DnG had a significantly larger path length than both CDM ($p = .009$) and DP ($p = .017$). This was the same for the horizontal ($p = .016$ and $p = .026$ respectively) and vertical tasks ($p < .001$ and $p = .022$ respectively). Additionally, CDM had a significantly shorter path than DP as well ($p = .043$). There were no statistically significant pairs for the long task.

Task	Time Mean (s)				Friedman Test or ANOVA
	CDR	C&C	GC	TSD	
Mixed	34.782 [a]	66.555 [a b c]	35.412 [b]	33.892 [c]	$F(1.418, 11.345) = 47.428, p < .001$
Small	30.906 [a]	49.950 [a b]	28.813	28.418 [b]	$\chi^2(3) = 11.914, p = .008$
Large	22.344 [a]	31.352 [a b c]	21.696 [b]	20.626 [c]	$F(3, 30) = 28.219, p < .001$

Table 6.3: *Orientation* technique times, [a b c] mean pair of statistic significance.

Going back to task times, but this time for *Orientation* tasks (table 6.3), the mixed *Orientation task* had the *Choose & Click* method perform worse than *Continuous Directional Rotation* ($p <$ .001), *Gaze Convergence* ($p < .001$), and *Tactile Surface Dragging* ($p < .001$). Same could be said for the large *Orientation* task ($p = .002$, $p < .001$ and $p < .001$ respectively). The small *Orientation* task only saw *C&C* performing worse than *CDR* ($Z = -2.981, p = .018$) and *TSD* ($Z = -2.934, p = .018$).

Task	Measurement Mean (º)				Friedman Test or ANOVA
	CDR	C&C	GC	TSD	
Mixed	615.147	656.624	547.144	581.240	$\chi^2(3) = 5.550, p = .136$
Small	193.103 [a]	212.363	176.083 [a]	190.760	$\chi^2(3) = 8.067, p = .045$
Large	498.490	599.737	493.760	528.701	$\chi^2(3) = 5.550, p = .136$

Table 6.4: *Orientation* technique measurements, [a] mean pair of statistic significance.

Finally, for the total rotation (table 6.4), the only statistically significant result of all three tasks was on the small *Orientation* task, where *Gaze Convergence* had a smaller total rotation than *Continuous Directional Rotation* ($Z = -2.803, p = .030$).

6.2 Questionnaire Results

Contrary to *User Performance*, the questionnaire data is discrete, therefore we will use Friedman Tests paired with a post-hoc Wilcoxon Signed-Rank. We still apply Bonferroni adjustments just like the previous results (3 possible pairs for 3 *Travel* methods, 6 possible pairs for 4 *Orientation* methods).

Questionnaire data can be divided into 3 sections, with a few questions to be addressed separately. First, we have self-assessment questions, where the user evaluates their performance with their current method, then symptomatology questions, and finally technique assessment questions, where the user evaluates how they feel about the method itself. We also asked participants to rank each method within their category, and rating of the final joint method.

To start off, all participants bar one rated the final joint method with a maximum grade of 5, with the exception giving a rating of 4, but only due to complaints of motion sickness when attempting to move and rotate at the same time.

As for technique rankings, we could not reach statistical significance with either *Travel* or *Orientation* techniques. For *Travel* techniques, *Continuous Directional Movement* had a median of 1.50 (IQR = 1), *Drag'n'Go*, 2.00 (IQR = 2), and *Dog Paddle*, 2.50 (IQR = 2). As for *Orientation* techniques, *Continuous Directional Rotation* had a median of 2.00 (IQR = 3), *Choose & Click*, 3.50 (IQR = 1), *Gaze Convergence*, 3.00 (IQR = 3), and *Tactile Surface Dragging*, 2.00 (IQR = 1).

Question	Medians (IQR)			Friedman Test
	CDM	*Drag'n'Go*	Dog Paddle	
How well do you think you performed?	$7(1)^a$	$5(1)^{a\ b}$	$6(3)^b$	$\chi^2(2) = 15.436, p < .001$
How much effort did it require to reach your level of performance?	$1(2)^a$	$3.5(2)^a$	3.5(4)	$\chi^2(2) = 7.171, p = .028$
How do you feel about your performance?	$7(1)^a$	$4.5(3)^a$	6(3)	$\chi^2(2) = 12.162, p = .002$
How MENTALLY demanding was the task?	$1(1)^a$	$3(3)^a$	2(3)	$\chi^2(2) = 11.118, p = .004$
How PHYSICALLY demanding was the task?	2(1)	2(2)	3.5(4)	$\chi^2(2) = 5.687, p = .058$
How rushed was your pace on the task?	1(3)	3(4)	3.5(5)	$\chi^2(2) = 7.000, p = .030$

Table 6.5: Performance Self-assessment for *Travel* techniques, ^{a} and ^{b} mean pair of statistic significance.

Moving on towards the individual questionnaires, participants believe that, in the *Travel* tasks (table 6.5), they performed worse with *DnG* than with *CDM* ($Z = -2.911, p = .012$) and *DP* ($Z = -2.652, p = .024$). *CDM* also required less effort than *DnG* ($Z = -2.491, p = .039$) to reach that level of performance. Users felt more confident with *CDM* than *DnG* ($Z = -2.701, p = .021$),

Question	Medians (IQR)			Friedman Test
	CDM	Drag'n'Go	Dog Paddle	
I would like to use this technique frequently.	4(1)	3(3)	3(3)	$\chi^2(2) = 7.257, p = .027$
I found the technique unnecessarily complex.	1(0)	1(1)	1.5(2)	$\chi^2(2) = 7.304, p = .026$
The technique was easy to use.	$5(0)^a$	$4(2)^a$	4(2)	$\chi^2(2) = 10.231, p = .006$
I think that I would need the support of a technical person to be able to use this technique.	1(1)	1(2)	1(1)	$\chi^2(2) = 3.364, p = .186$
I would imagine that most people would learn to use this technique very quickly.	$5(1)^a$	$4(2)^a$	4(2)	$\chi^2(2) = 8.424, p = .015$
I found the technique very cumbersome to use.	1(1)	2(2)	2(3)	$\chi^2(2) = 5.643, p = .060$
I felt very confident using the technique.	5(1)	4(2)	4(2)	$\chi^2(2) = 7.818, p = .020$
I needed to learn a lot of things before I could get going with this technique.	1(0)	1(1)	1(1)	$\chi^2(2) = 3.818, p = .148$

Table 6.6: Travel technique assessment, ^{a} mean pair of statistic significance.

and also found the first to be less mentally demanding than the latter ($Z = -2.694, p = .021$). They also felt their pace was more rushed with *DP* than with *CDM* ($Z = -2.414, p = .048$).

As for technique assessment (table 6.6), participants found *CDM* easier to use than *DnG* ($Z = -2.598, p = .027$), as well as easier to learn ($Z = -2.598, p = .027$).

Question	Medians (IQR)				Friedman Test
	CDR	C&C	GC	TSD	
How well do you think you performed?	$6.5(1)^a$	$4.5(2)^{a\ b\ c}$	$6(1)^b$	$6.5(1)^c$	$\chi^2(3) = 25.041, p < .001$
How much effort did it require to reach your level of performance?	$1(2)^a$	$5(2)^{a\ b}$	3(4)	$2(2)^b$	$\chi^2(3) = 21.088, p < .001$
How do you feel about your performance?	$7(1)^a$	$3(3)^{a\ b}$	6(2)	$7(1)^b$	$\chi^2(3) = 20.194, p < .001$
How MENTALLY demanding was the task?	$1(1)^a$	$3.5(3)^{a\ b}$	2(3)	$1.5(1)^b$	$\chi^2(3) = 14.798, p = .002$
How PHYSICALLY demanding was the task?	$1(1)^a$	2.5(2)	$3(4)^a$	1.5(2)	$\chi^2(3) = 9.620, p = .022$
How rushed was your pace on the task?	1(4)	4(2)	4.5(4)	2(4)	$\chi^2(3) = 3.282, p = .350$

Table 6.7: Performance Self-assessment for *Orientation* techniques, $^{a\ b\ c}$ mean pair of statistic significance.

As for *Orientation* techniques (table 6.7), participants said they performed significantly worse with *C&C* than with *CDR* ($Z = -2.988, p = .018$), *GC* ($Z = -2.676, p = .0.42$), and *TSD* ($Z = -3.130, p = .012$). *C&C*, compared to *CDR* and *TSD*, also required more effort ($Z = -3.084, p = .012$ and $Z = -2.848, p = .024$ respectively), left the users less confident ($Z = -2.675, p = .042$ and $Z = -3.048, p = .012$ respectively), and was more mentally demanding ($Z = -2.680, p = .042$ and $Z = -2.701, p = .042$ respectively). *GC* was significantly more physically demanding than *CDR*.

Continuing with technique assessment (table 6.8), *CDR* and *TSD* appeared to be significantly easier than *C&C* ($Z = -2.971, p = .018$ and $Z = -3.097, p = .012$ respectively). Participants also evaluated *TSD* as being more confidence inspiring than *C&C*.

Question	Medians (IQR)				Friedman Test
	CDR	C&C	GC	TSD	
I would like to use this technique frequently.	5(2)	3(2)	4.5(3)	4(3)	$\chi^2(3) = 4.598, p = .204$
I found the technique unnecessarily complex.	1(0)	2.5(3)	1(1)	1(0)	$\chi^2(3) = 11.081, p = .011$
The technique was easy to use.	$5(0)^a$	$3(2)^{a\ b}$	5(2)	$5(0)^b$	$\chi^2(3) = 21.233, p < .001$
I think that I would need the support of a technical person to be able to use this technique.	1(0)	1(1)	1(1)	1(0)	$\chi^2(3) = 2.800, p = .423$
I would imagine that most people would learn to use this technique very quickly.	5(0)	3(3)	4(2)	5(1)	$\chi^2(3) = 13.963, p = .003$
I found the technique very cumbersome to use.	1(1)	3(4)	*1.5(3)*	1(0)	$\chi^2(3) = 6.556, p = .087$
I felt very confident using the technique.	5(0)	$3(2)^a$	4.5(2)	$5(0)^a$	$\chi^2(3) = 16.735, p < .001$
I needed to learn a lot of things before I could get going with this technique.	1(0)	1(2)	1(1)	1(0)	$\chi^2(3) = 9.588, p = .022$

Table 6.8: *Orientation* technique assessment, $^{a\ b}$ mean pair of statistic significance.

Figure 6.1: Symptomatology for *Travel* tasks

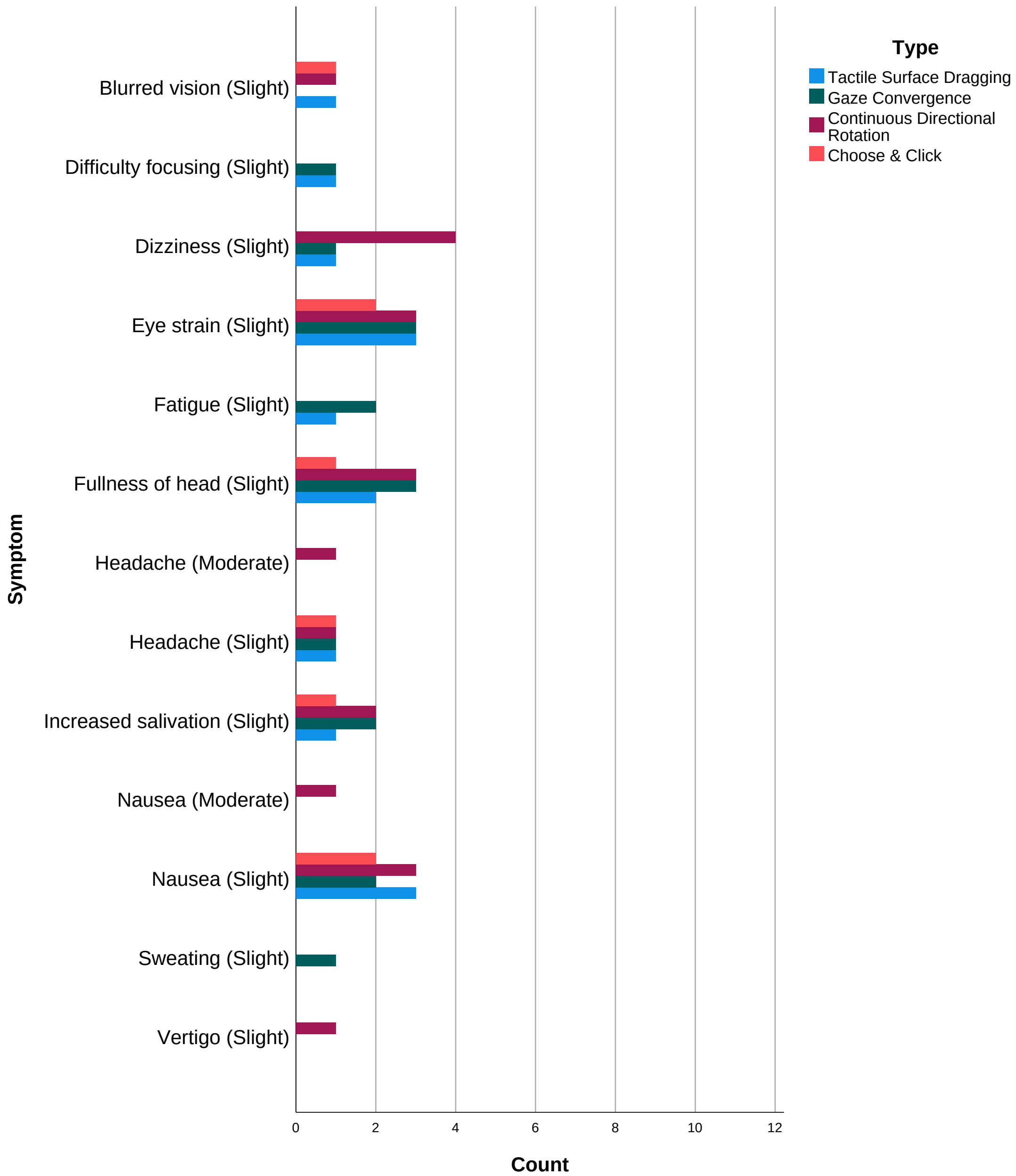

Figure 6.2: Symptomatology for *Orientation* tasks

6.3 Observations

During each participation, observations were made to user behavior, complaints and prototype issues.

Starting off with the application itself, one of the methods proved to be problematic. During testing, the *Drag'n'Go* method caused the prototype to crash a total of 4 times. There were also multiple occurrences of participants *phasing* through buildings when reversing. Despite this, there were only 2 occasions where the task had to be retaken.

Additionally, during the final test, where participants choose their favorite technique from each category, the *Drag'n'Go* method would interrupt rotation if the chosen *Orientation* method was *Tactile Surface Dragging*. To overcome this, during the final section, each method would be turned on and off as the participant requested in order to try the junction of the two.

As far as hardware issues, there was 1 recorded incident of the the hand controller tracking suddenly failing, and nothing short of a complete system reboot fixed it. As this was tested in between tasks, it caused no significant issues. There were also issues with the finger and hand tracking of the touch surface. As it uses infrared sensors to detect an interruption of a beam, it had issues when the user did not lift their hand high enough, or fingers came too close to each other, more so when both hands came near one another. This was particularly frustrating during *Dog Paddle* tasks, as it would often act as if both hands were only one, causing a sudden jolt in the opposite direction of movement.

Continuing with *Dog Paddle* and *Drag'n'Go*, a shared behavior across all participants was that they instinctively used a very small portion of the touch surface. Even with *Drag'n'Go*, where they could benefit from a higher degree of precision, they appeared to favor short, quick motions. In addition to this, users had a tendency to not aim at any object in particular, or at the checkpoint, or at the floor when using *Drag'n'Go*. The exception was during the last *Travel* task. They would often plan ahead and purposefully aim at far away objects, with varying degrees of efficiency.

There were a few complaints around the gaze-dependent techniques. 1 user reported slight neck pain during the vertical task with both *Dog Paddle* and *Drag'n'Go* techniques, as well as when using *Gaze Convergence* for larger angles. There were also other users who showed discomfort during gaze-oriented techniques, either physical or cybersickness. Another recurring complaint about *Gaze Convergence* was counter-rotation after applying the new orientation. At least 2 participants mentioned becoming disoriented when returning to center, but not while rotating. The same 2 participants did not have the same complaint with *Choose & Click*, which uses a very similar rotation system.

Finally, there were some complaints about the *Dog Paddle* input re-orientation based on the gaze's own orientation. 5 people said it was unintuitive. In addition, some users tried to move perpendicularly to the gaze direction, which was not possible.

One particular user which knew suffered from vertigo claimed that they only started noticing that symptom after realizing they were not experiencing vertigo even after a full set of tasks, which include vertical and flight sections over building roofs.

Overall, most tests lasted between 45 minutes to an hour. This was not timed, just a post-experiment observation.

6.4 Discussion

As far as ease-of-use, efficiency and convenience goes, *Continuous Directional Movement* had a clear advantage over its competitors. In mixed, horizontal and vertical tasks, it showed a statistically significant time and path length advantage over *Drag'n'Go*, and an advantage trend over *Dog Paddle*, even as far as statistically better in the vertical task. Also, it does not suffer from neck related issues, as it uses the hand rather than the head to establish a direction of movement. It also fares well in user evaluation, often doing better than *Drag'n'Go*, and trending slightly above *Tactile Surface Dragging*. The major issue it had in performance was that, unlike the other methods which allow to go as fast as you can swipe, *CDM* was limited to the speed limit set at full-trigger pressure, and possibly a conservative one at that. There were suggestions from some participants to add an extra button to allow for a "boost", increasing the maximum speed of travel. However, it should be noted that this would be a function extension. Any method given a functional extension would likely improve their results at their worse performances, and that would hide shortcomings of each technique. Additionally, each method could have multiple possible extensions, greatly increasing the final workload and testing time.

As for rotation, it was not as clear cut. Except for *Choose & Click*, all did reasonably the same, performance-wise. However, **there is a trend towards *Tactile Surface Dragging* as the best method overall**. It shares the user score trend from *Continuous Directional Rotation* and efficiency trend of *Gaze Convergence*, but is not statistically significantly better than its counterparts.

Moving towards symptomatology, it's hard to obtain conclusive answers. 2 participants did not experience any symptom during all tasks and tests. Only 1 method in each category experienced Moderate (or higher) symptoms, and only once each. **The onset of symptoms was overall low, and some were hard to distinguish, making observation error-prone.** Even more interestingly, some users reported that some symptoms only showed up during their first techniques, and would disappear even before a full set of tasks was complete. Still, some trends could be observed. Both *Drag'n'Go* and *Continuous Directional Rotation* were the only ones to experience moderate symptoms in each of their categories. In addition to this, *CDR* fared noticeably worse with Dizziness compared to the remaining methods. In the end, we should also take into account the fact that most users had little to no VR experience.

One interesting point that came up during testing was vertigo. 2 users claimed to have issues in real life with vertigo, and yet, they experienced it sparingly. More interestingly, one claimed they had not felt vertigo until mid way. This gave us an idea, as to whether **having the user focused on the task could alleviate cybersickness.** *CDM* has a slight trend of worse cybersickness than *DP*, but generally better usability. At the same time, *C&C* required a lot of precision and focus to get the correct orientation to complete the checkpoints, but generally lower symptom occurrence, despite significantly higher task duration.

In addition to this, **we speculate that direct-action inputs may reduce the onset of symptoms.** For example, *Dog Paddle* has noticeably lower number of occurrences of slight nausea compared to *CDM*, despite the movement itself being similar but jerkier. The same comparison can be made between *Continuous Directional Rotation* and *Tactile Surface Dragging*, but for Dizziness and other symptoms in general. It is possible that, by having the user expect some sort of visual response directly from an action, it may reduce the severity of symptoms.

One problem that became apparent about the *Orientation* tasks was the excessive focus on precision. While important, the main objective of these tasks should be to evaluate the ability to re-orientate the user's virtual body, not to aim it in a very specific direction. As such, certain techniques might have suffered excessively in time and questionnaire evaluations. This was the case of *Choose & Click*, where many users lost a significant chunk of time lining up the body with the checkpoint within the 2º of tolerance. The analog stick may be too small and the thumb control too imprecise to guide orientation with such precision. However, it might still be sufficient for most normal exploration scenarios.

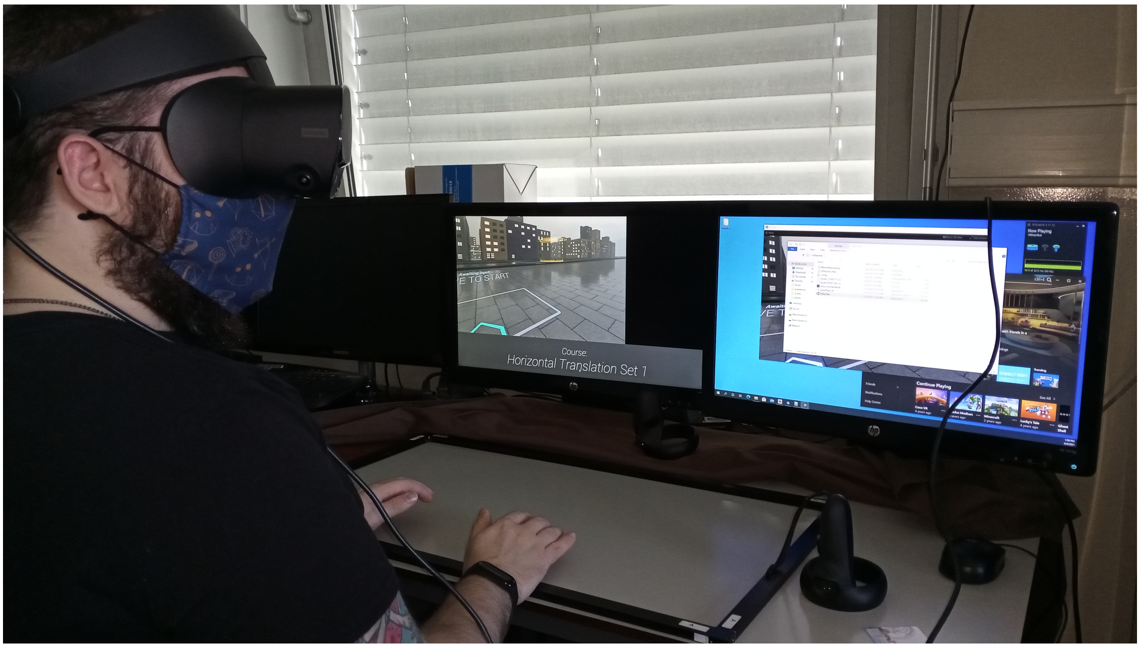

Figure 6.3: Example of participant only using small portion of available space

Another problem that quickly surfaced was that the touch surface used was sub-par for the given tasks. The infrared system it uses is not the most accurate, leading to tracking issues, but there was also other mishaps, such as foreign objects being detected as contacts and imperfections. While those could still be controlled, the tracking issues could not. This may have negatively biased participant grades and preferences, due to the jitteriness of the movement in certain occasions. This could be remedied somewhat with the use of a single finger per hand, and by spreading the hands apart, but this still counteracted the user's intuitive interaction. Even though the surface occupied a significant portion of the working table, the tendency was to use a small area close to the body, with small but short movements.

6.5 Summary

In this chapter, we reported the data obtained and possible findings. We found a clear preference for *Continuous Directional Movement* in terms of ease-of-use and overall performance, and a slight trend towards *Tactile Surface Dragging* for *Orientation*. We also speculate that direct-action input and task focus may reduce the onset of symptoms. Overall symptomatology was reduced, with 2 participants not reporting any symptom during the entire evaluation. We also mention some notes regarding the poor accuracy of the touch frame, and the possibly unnecessary precision of the *Orientation* tasks.

Chapter 7

Conclusions

As we look into the future of VR, it is possible that it may become a multi-purpose tool for development of other different areas, such as current desktop computers. But just like desktop computers, desktop VR, or more broadly, seated VR limits the type of interactions we can have with the environment, which, in this case, is virtual.

We looked at current solutions for movement techniques that do not require physical movement and proposed some of our own. We then produced a prototype and test methodology to compare and obtain information. The results did not give a direct answer to which method was best, nor a clear preference to a specific technique, but nevertheless, we observed a few points of interest that could be looked into.

The experiment itself was also well-received. The onset of symptoms was reduced, and all participants were happy with the resulting preferred joint method. This was notably important, as a bad first experience can keep possible interested parties away from the VR ecosystem.

7.1 Future Work

From the results obtained, we have identified a few key areas that may elicit future work. For example, further developing a control scheme that includes both *Continuous Directional Movement* and *Tactile Surface Dragging* could prove successful. Additionally, supplemental study of the effects of task focus on cybersickness could help mitigate these issues during normal use. Finally, direct-action input could also have a significant impact on motion sickness, but it will require additional investigation.

References

[1] Ferran Argelaguet and Morgant Maignant. GiAnt: Stereoscopic-compliant multi-scale navigation in VEs. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, 02-04-Nove:269–277, 2016.

[2] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. Performance and Ergonomics of Touch Surfaces. pages 1817–1826, 2015.

[3] Rui Miguel de Paiva Batista. Navigating Virtual Reality Worlds with Leap Motion Controller. 2016.

[4] B Bolte, G Bruder, and F Steinicke. The Jumper Metaphor: An Effective Navigation Technique for Immersive Display Setups. *Proceedings of Virtual Reality International Conference (VRIC)*, (April):6–8, 2011.

[5] Fabio Buttussi and Luca Chittaro. Locomotion in place in virtual reality: A comparative evaluation of joystick, teleport, and leaning. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):125–136, 2020.

[6] Sebastian Cmentowski, Andrey Krekhov, and Jens Krüger. Outstanding: A multi-perspective travel approach for virtual reality games. *CHI PLAY 2019 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 287–299, 2019.

[7] Tiare Feuchtner. Designing for Hand Ownership in Interaction with Virtual and Augmented Reality. (February), 2020.

[8] Jann Philipp Freiwald, Oscar Ariza, Omar Janeh, and Frank Steinicke. Walking by Cycling: A Novel In-Place Locomotion User Interface for Seated Virtual Reality Experiences. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–12, 2020.

[9] Markus Funk, Florian Müller, Marco Fendrich, Megan Shene, Moritz Kolvenbach, Niclas Dobbertin, Sebastian Günther, and Max Mühlhäuser. Assessing the Accuracy of Point & Teleport Locomotion with Orientation Indication for Virtual Reality using Curved Trajectories. *Conference on Human Factors in Computing Systems - Proceedings*, (Chi):1–12, 2019.

[10] M. P. Jacob Habgood, David Moore, David Wilson, and Sergio Alapont. Rapid, Continuous Movement between Nodes as an Accessible Virtual Reality Locomotion Technique. *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, pages 371–378, 2018.

[11] M. P. Jacob Habgood, David Wilson, David Moore, and Sergio Alapont. HCI lessons from PlayStation VR. *CHI PLAY 2017 Extended Abstracts - Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, pages 125–135, 2017.

[12] Ji Sun Kim, Denis Gračanin, Krešimir Matković, and Francis Quek. Finger Walking in Place (FWIP): A traveling technique in virtual environments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5166 LNCS(August):58–69, 2008.

[13] Ji Sun Kim, Denis Gračanin, Taeyoung Yang, and Francis Quek. Action-transferred navigation technique design approach supporting human spatial learning. *ACM Transactions on Computer-Human Interaction*, 22(6), 2015.

[14] Andrey Krekhov, Sebastian Cmentowski, Katharina Emmerich, Maic Masuch, and Jens Krüger. GulliVR: A walking-oriented technique for navigation in virtual reality games based on virtual body resizing. *CHI PLAY 2018 - Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, pages 243–256, 2018.

[15] Daniel Medeiros, Eduardo Cordeiro, Daniel Mendes, Maurício Sousa, Alberto Raposo, Alfredo Ferreira, and Joaquim Jorge. Effects of speed and transitions on target-based travel techniques. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, 02-04-Nove:327–328, 2016.

[16] Daniel Medeiros, Mauricio Sousa, Alberto Raposo, and Joaquim Jorge. Magic Carpet: Interaction Fidelity for Flying in VR. *IEEE Transactions on Visualization and Computer Graphics*, 26(9):2793–2804, 2020.

[17] Daniel Mendes, Maurício Sousa, Alfredo Ferreira, and Joaquim Jorge. ThumbCam: Returning to single touch interactions to explore 3D virtual environments. *ITS 2014 - Proceedings of the 2014 ACM International Conference on Interactive Tabletops and Surfaces*, pages 403–408, 2014.

[18] Kasra Moghadam, Colin Banigan, and Eric D. Ragan. Scene Transitions and Teleportation in Virtual Reality and the Implications for Spatial Awareness and Sickness. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2273–2287, 2020.

[19] Daniel Zielasko, Benjamin Weyers, and Torsten W. Kuhlen. A non-stationary office desk substitution for desk-based and HMD-projected virtual reality. *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, (February):1884–1889, 2019.