

Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Diseño y Desarrollo de Videojuegos

Proyecto Final

**Diseño e Implementación de un Sistema de
Navegación en Entornos de Realidad Virtual
para Videojuegos Comerciales**

Autor del proyecto: Sergio Jimeno Navarro

Director del proyecto: Daniel Blasco Latorre

Zaragoza, 11 de Septiembre de 2020



Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Diseño y Desarrollo de Videojuegos por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma

Fecha

A handwritten signature in black ink, appearing to be 'G. Lopez', written in a cursive style.

11 de Septiembre de 2020

Dedicatoria y Agradecimiento

Quiero dedicarle este proyecto a mucha gente que ha aportado su granito de arena en esto, directa o indirectamente.

A mi familia por apoyarme durante toda la carrera y por haberme dejado a mi aire mientras realizaba el proyecto.

A mi pareja, Isa, por aguantarme todos estos años, apoyarme en absolutamente todo lo que me propongo y por ayudarme a mejorar.

A mis amigos de Entalto Studios y 4FreaksFiction, aquí nació la idea y de aquí surgirán muchas más.

Al profesorado de la Universidad San Jorge y a la Universidad en general. Gracias en especial a Daniel Blasco y Antonio Iglesias por todas la ayuda dentro y fuera de clase que me habéis dado, y gracias también a Jesús Carro, que me dejaba las gafas de realidad virtual siempre que se lo pedía.

A mis quince testers, gracias por sacar tiempo para ayudarme, sin vosotros no habría podido hacerlo.

Table of Contents

Resumen	1
Abstract.....	1
1. Introduction	3
2. State of the art	5
2.1. Motion Sickness	5
2.2. Locomotion.....	6
<i>2.2.1. External Controller based locomotion.....</i>	<i>6</i>
<i>2.2.2. Headset/Hand Controller based locomotion.....</i>	<i>7</i>
2.3. Personal Experience	13
3. Objectives.....	17
4. Methodology.....	19
4.1. Study Phases	19
4.2. Working methodology	20
4.3. Project structure	21
5. Economic study	23
6. Study Development	27
6.1. Summary	27
6.2. Covid-19 measures.....	28
6.3. Testing subjects and questionnaire.	29
6.4. Unity and its tools	30
6.5. Environment.....	32
6.6. Device and key mapping	35
6.7. Locomotion Implementation.....	36
<i>6.7.1. Player Hierarchy.....</i>	<i>36</i>
<i>6.7.2. Raw, Reduced FOV and Static Reference</i>	<i>38</i>
<i>6.7.3. Teleport.....</i>	<i>41</i>
<i>6.7.4. Smooth Continuous</i>	<i>43</i>
<i>6.7.5. Arm Swinging.....</i>	<i>43</i>
<i>6.7.6. Flying</i>	<i>44</i>
6.8. Development of the testing	47
7. Results.....	49
7.1. Teleportation.....	51
7.2. Smooth Continuous.....	54
7.3. Arm Swinging	56
7.4. Flying.....	59
8. Conclusions	65
Annex I – Questionnaire	67
Annex II – Tests Results:.....	69
Annex III – Propuesta TFG:	71
Annex IV - Tutorías	72

Bibliography 73

Table of Figures

Figure 1: VR Software Sales by Platform (2017-2020) 3

Figure 2: Global VR headset market share, by application 4

Figure 3: Omni Platform (Left) and Vive Trackers (Right) 7

Figure 4: FOV in anamorphic video game in 16:9 resolution (Left) and 4:3 resolution (Right) .. 10

Figure 5: Different headsets and humans FOV 10

Figure 6: Most common VR locomotion types by percentage 12

Figure 7: VR Locomotion "Family Tree" 13

Figure 8: Screenshot of the video game HAT 14

Figure 9: Project process..... 17

Figure 10: Pre-Study Phase Timeline..... 19

Figure 11: The target..... 28

Figure 12: Covid-19 security measures..... 29

Figure 13: The testing scenario 33

Figure 14: Screenshots of the testing scenario populated and illuminated. 34

Figure 15: Vive Cosmos 35

Figure 16: Vive Cosmos controllers – Joysticks (1, left side) – Triggers (2, right side) 35

Figure 17: Original Player Hierarchy (Left) – Player Teleport with Static Reference (Right) 36

Figure 18: Player using Raw variant..... 38

Figure 19: Players view using Raw variant..... 38

Figure 20: Player using Reduced FOV variant 39

Figure 21: Players view using Reduced FOV variant 39

Figure 22: Player using Static Reference variant 40

Figure 23: Players view using Static Reference variant 40

Figure 24: Valid teleport objective 41

Figure 25: Invalid teleport objective..... 42

Figure 26: Thruster position and rotation in each hand 45

Figure 27: Addition of two vectors 46

Figure 28: Graph example 50

Figure 29: Teleport system graph without reduced FOV or Static Reference 51

Figure 30: Teleport system graph with reduced FOV 52

Figure 31: Teleport system graph with Static Reference 53

Figure 32: Smooth Continuous system graph without reduced FOV or Static Reference	54
Figure 33: Smooth Continuous system graph with reduced FOV	55
Figure 34: Smooth Continuous system graph with Static Reference	55
Figure 35: Arm Swinging system graph without reduced FOV or Static Reference	56
Figure 36: Arm Swinging system graph with reduced FOV	57
Figure 37: Arm Swinging system graph with Static Reference	58
Figure 38: Flying system graph without reduced FOV or Static Reference	59
Figure 39: Flying system graph with reduced FOV	60
Figure 40: Flying system graph with Static Reference	61
Figure 41: Percentage of subjects with optimal response	62
Figure 42: Percentage of subjects with worst response	62

Table of Tables

Table 1: Most sold VR games and used locomotion types	6
Table 2: Total cost of the study	25

Table of Abbreviations and Acronyms

VR: Virtual Reality

GTD: Get Things Done

FOV: Field Of View

SR: Static Reference

PC: Personal Computer

NPC: Non Playable Character

Resumen

La cada vez mayor relevancia de los dispositivos de realidad virtual en entornos profesionales y domésticos evidencia la necesidad de mejorar los actuales sistemas de control y crear soluciones más eficaces y mejor adaptadas al ser humano.

Este proyecto se centra en la creación de un sistema de navegación que, mediante el análisis de aproximaciones previas y de las necesidades de los usuarios en el entorno práctico de los salones recreativos, sirva para estudiar posibles vías de mejora en cuestiones como la precisión, la percepción espacial y el equilibrio.

Los resultados intentarán demostrar como los distintos tipos de locomoción que se han probado en este proyecto afectan a los usuarios, si alguna de sus variantes o alternativas reducen la cinetosis y de como la creación de nuevos sistemas de locomoción pueden combatirla al mismo tiempo que expresen el potencial oculto de esta tecnología.

Abstract

The increasing relevance of virtual reality devices in professional and domestic environments shows the need to improve current control systems and create more effective solutions better adapted to the human being.

This project focuses on the creation of a navigation system that, by analysing previous approaches and the needs of users in the practical environment of arcades, serves to study possible ways of improvement in matters such as precision, spatial perception and balance.

The results will attempt to demonstrate how the different types of locomotion that have been tested in this project affect users, if any of its variants or alternatives reduce motion sickness and how the creation of new locomotion systems can combat it at the same time that they squeeze the hidden potential of this technology.

1. Introduction

When we hear the words “Virtual Reality” the first thing it usually pops in everybody’s mind is us, inside a virtual world with endless possibilities, which we can not only see, but explore, touch, and even feel. Books, series and movies like *Ready Player One* (2018), have shown us the potential this technology have, and what marvels it can achieve in the future.

In the last years, VR headsets have been appearing and expanding around the market, made by Tech giants like *Facebook* and *Sony* among others. With each headset, a new improvement comes along: better resolution, less weight, wireless technology, No-PC support...

Although the Covid-19 global crisis will surely affect sales for a time, virtual reality devices and video games have been growing in sales since 2017, talking in the likes of millions of dollars of benefit.

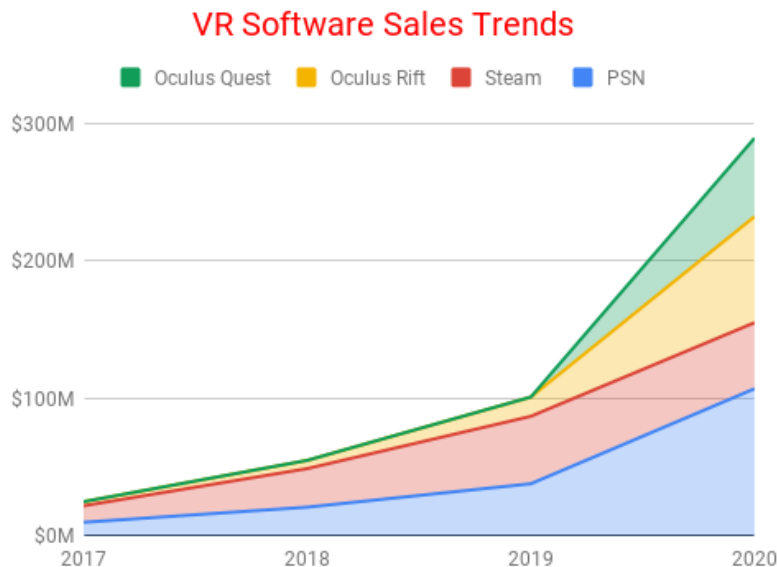


Figure 1: VR Software Sales by Platform (2017-2020)

This growth have been possible thanks to the gaming industry. Even though VR technology has other applications like education, media and entertainment or even military training, the gaming industry is the one that is currently taking the most advantage of it and drives the evolution of this technology. With thousands of games and experiences on the market the industry and market are doing nothing else but grow.

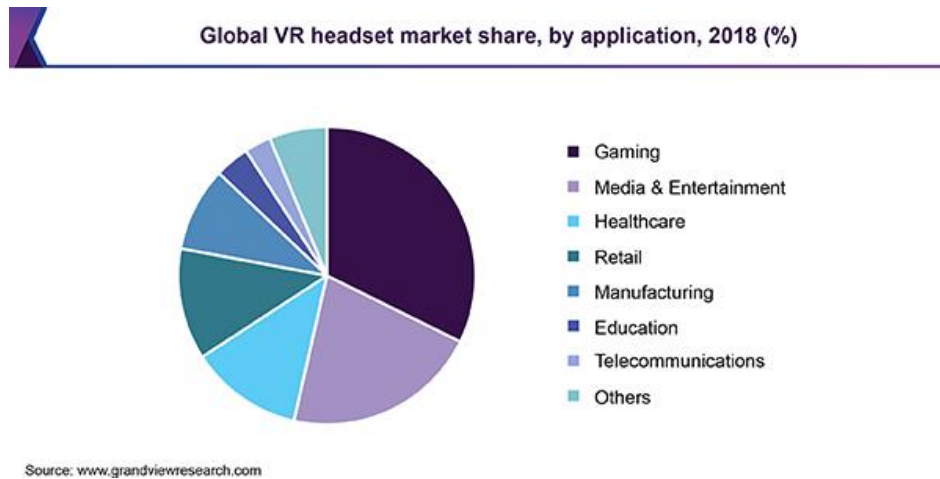


Figure 2: Global VR headset market share, by application

So, can we do it? Can we go inside a magical world and do all those things we have seen in the movies? Yes and no. The technology is still in its early stages, we can see and touch thanks to our headsets and controllers, and hopefully, with new devices like *Omni Platform* [1] (a VR motion platform) we will be able to move inside any game by using our own legs.

So, if we are not moving with our legs, how are we moving at all? Depends on the game. Locomotion in VR is a tricky business, mainly because of what we call "Motion Sickness", that infamous dizziness that some people get when using VR. Because of this, video games developers are really careful when they work on a VR title, because they don't want to create a product where their players might end up sick and therefore losing money, so they usually resort to implement a teleport system or not implementing any type of locomotion at all.

I think this is a huge mistake, VR technology have a lot of cons already like its price, the necessity of owning a compatible PC or a big enough room to play in, and now with the Covid-19 crisis both the VR and gaming industries are going to take a toll. It is the job of developers to experiment with different ways to allow the player to move in their games and fight the stigma of motion sickness, offering the public experiences that actually feel like those movies we have seen, and therefore, allowing the industry to grow and evolve to its full potential.

In this essay I will try to find answers to the mystery of motion sickness, try to find which types of movement are more linked to it and, at the same time, try to find how to make the players have fun and not just stand around.

2. State of the art

2.1. Motion Sickness

Motion sickness is one of the biggest problems virtual reality is facing right now. Motion sickness happens when there is a difference between the motion we are perceiving and the one that is actually occurring, which can induce to nausea, vomiting, dizziness, headaches and many other symptoms. This is the reason both players and developers “fear” virtual reality, nobody wants to suffer when they are supposed to have fun.

Right now there is not a single theory on what causes motion sickness, but there are a couple which would explain on why VR can induce it.

The first would be the Defence against Poisoning Theory [2]. This theory holds that motion sickness is a defence mechanism against neurotoxins. When an individual consumes a poison or other substance with neurotoxins, the body will suffer as a result of said intoxication, which can cause conflicts between the sense of sight and that of balance. To resolve this conflict, the brain induces vomiting to rid itself of the toxic agent. The most common example for this theory would be alcohol poisoning. When we consume alcohol, the part of the head responsible for regulating the balance is drastically affected, causing a balance-vision conflict, and therefore the consequent nausea and vomiting which tries to get rid of neurotoxins.

The second theory, and one that seems to be more suitable for our study, is the Sensory Conflict Theory [3]. This theory argues that when the brain receives two incongruous states of locomotion, the result is nausea and dizziness. At first glance it may seem that this theory is the same as the previous one, and in a way the basis could be the same, the main difference is that the cause of this new theory is a discrepancy between what the body sees and what it feels because they are definitely different, and the previous one attributes it to external agents that affect the biology of the human body. Although the individual is not being intoxicated, the body may believe that it is, and therefore the result is the same for both theories, dizziness and nausea.

2.2. Locomotion

When we talk about VR locomotion, I would divide it in at least two big categories: Locomotion achieved by using hand controllers or our headset as the first category, and locomotion with external controllers as the second. This difference is pretty important, not only for the player but for the developers too.

The first type is pretty self-explanatory, the movement (if any) is achieved and controlled by the use of the headset or the hand controllers that are included with it. It is the most common one and it is divided in different types of locomotion which we will explain later, first let's talk about external controller based locomotion.

2.2.1. External Controller based locomotion

In this category the player requires the use of an external controller or device to move, devices like external trackers or motion platforms. This type of devices are not usually sold together with the headsets because they are the product of tech companies which create and produce them separately as their own product, like the *Omni Platform*. Sometimes the companies launch extra devices in an attempt to put them in front of other headset companies, and create products like the *Vive Trackers* [4].

It doesn't matter where the product comes from, but its purpose. The motion platforms allows the players to walk, run, crouch and even jump in real life, avoiding motion sickness [5] and improving immersion. Trackers can be used to track props like weapons or other objects, but they can also be used to track your own feet, allowing the developers to create locomotion systems by tracking the player's feet.

The downside unfortunately is the price, both options are usually too expensive and only a few players ever buy any of these products besides the headset, for this reason, developers do not usually spend time creating games which support any of this options.



Figure 3: Omni Platform (Left) and Vive Trackers (Right)

2.2.2. Headset/Hand Controller based locomotion

As we mentioned, the most common types of locomotion are the ones we achieve by either the use/movement of our headset or hand controllers. Up to the date, there are several types of locomotion systems in VR and each developer chooses one or another depending on the type of experience they are creating.

To get an idea, I investigated and selected thirty of the most selling VR videogames to this moment and compiled which type of movement each one of them employ.

	Still	Room	Rail	Teleport	Teleport (Blink)	Teleport (Dash)	Continuous Smooth	Continuous Smooth (Fov)	Arm Swinging/Head Movement	LINK
Half-Life: Alyx					X	X	X			https://www.youtube.com/watch?v=TX58AbJq-xo
The Walking Dead: Saints & Sinners							X	X		https://www.youtube.com/watch?v=z-3dVBP49jA
DreadEye VR	X		X							https://www.youtube.com/watch?v=ABb1bAAAb58s
Arizona Sunshine					X		X			https://www.youtube.com/watch?v=E5LzC-g0C5s
AUDICA	X									https://www.youtube.com/watch?v=ifdHueVHJHM
Stand Out			X				X		X	https://www.youtube.com/watch?v=kJ6c0RsWWJA
War Dust			X				X		X	https://www.youtube.com/watch?v=0z-JVVHKAgk&t=96s

Beat Saber	X									https://www.youtube.com/watch?v=3zTkxuqy4g
Hellsplit: Arena								X		https://www.youtube.com/watch?v=TUVArVy4zI0
Pavlov VR								X		https://www.youtube.com/watch?v=3LTNwwA8csQ&t
Red Matter						X				https://www.youtube.com/watch?v=zzGULMdwCrQ
Contractors								X		https://www.youtube.com/watch?v=BcZpyoRTCmY
BONEWORKS								X		https://www.youtube.com/watch?v=q70Ne40h57U
Pistol Whip			X							https://www.youtube.com/watch?v=9sIUHh9CrI
Blade and Sorcery								X		https://www.youtube.com/watch?v=5wx7OomJNS4
Surgeon Simulator: Experience Reality	X	X								https://www.youtube.com/watch?v=xn-3g9YdFqk
Elven Assassin				X						https://www.youtube.com/watch?v=rh0nLLgqvVM
Ocean Rift	X							X		https://www.youtube.com/watch?v=uOdEu1z72QQ
Synth Ryders			X							https://www.youtube.com/watch?v=7nj18rdMIWc
Catan VR	X									https://www.youtube.com/watch?v=NEjjOqWpnU8
Dreadhalls								X		https://www.youtube.com/watch?v=ct1_YTMBaU0
Guns'n'Stories: Bulletproof VR	X									https://www.youtube.com/watch?v=YP6jQeEtmjs
SUPERHOT VR	X									https://www.youtube.com/watch?v=mhR8boRzXnI
Gorn								X	X	https://www.youtube.com/watch?v=4nsg3t573oI
Skyrim VR					X				X	https://www.youtube.com/watch?v=ZXTmSejMEk
Hot Dogs, Horseshoes & Hand Grenades								X		https://www.youtube.com/watch?v=4Q0aV1JRqpY
Zero Caliber								X		https://www.youtube.com/watch?v=ayoSs7p9bkM
Fallout 4						X		X		https://www.youtube.com/watch?v=S0DON2SYHII
Job Simulator	X	X								https://www.youtube.com/watch?v=k6oWdMWj8oU
Robo Recall					X					https://www.youtube.com/watch?v=jgu4VmiMSGw

Table 1: Most sold VR games and used locomotion types

We can see from this table that there are about five types of locomotion, the most common ones. We can find other types of course, more experimental, but they are not used in enough games to be considered as regular. The types are the following:

None: No locomotion at all, the player keeps still in place through the game or needs to move only a few centimetres inside the tracking zone. This method does not usually create Motion sickness in players, that's why a long range of games, specially the first ones to appear on the market, use this type.

Rail: The player is moving without any kind of input from him, maybe because the player is in some kind of vehicle that moves by itself or any other similar method. This was one of the attempts to implement locomotion while avoiding motion sickness, unfortunately the result is the same as the one that happens in real life, if you are prompt to get dizzy in a car or a boat you will most surely get dizzy using this method.

Teleport: This is believed to be the most popular type of locomotion system. The player will press a button on its controller and will be teleported to another location. This teleport can be regular, with some short of blink in which the screen turns black (or other colour) or instead of a literal teleportation it can be a dash, a quick movement to the target point. Except for the dash variant, teleportation is believed to be the best locomotion method to avoid motion sickness, this is only because the player does not "see" the movement, he just goes from one point to the next in an instant, like the blink of an eye, which is the reason the blink variant of teleportation exists.

Continuous Smooth: This type of locomotion is the same regular video games employ to move the player, which means to use a joystick, thumpad or a series of buttons to make the player move gradually at some speed through the scenario. This is believed to be one of the worst type of movements due to the high number of players that usually suffers from motion sickness using it. For that reason, it can also feature what it is called "Reduced FOV". Some headsets have a lower field of view already trying to achieve this.

Before continuing let's dig in what we mean by Field Of View. The Field of View in videogames is, by definition, "the extent of the observable game world that is seen on the display at any given moment" [6].

It is important to explain that the FOV on regular videogames is not the same as we would use in VR. For other video games the FOV depends of the resolution of the monitor you are playing and the image will change depending on the type of FOV method the game uses. For example: If you use the *Hor+* method, the FOV will increases with bigger resolutions, which means you'll see more parts of the scene. On the other hand, if you use the anamorphic method, the image will keep its ratio and it will be letterboxed [7].



Figure 4: FOV in anamorphic video game in 16:9 resolution (Left) and 4:3 resolution (Right)

In VR this methods can't be implemented, after all, one of the key elements of VR is to make you feel like you are inside the game, not spectating it. Human vision has between 200 and 220 degrees of view, unfortunately VR headset can't reach that number of degrees with the current technology, even though we are slowly getting there with each new headset that comes to market (Figure 5).

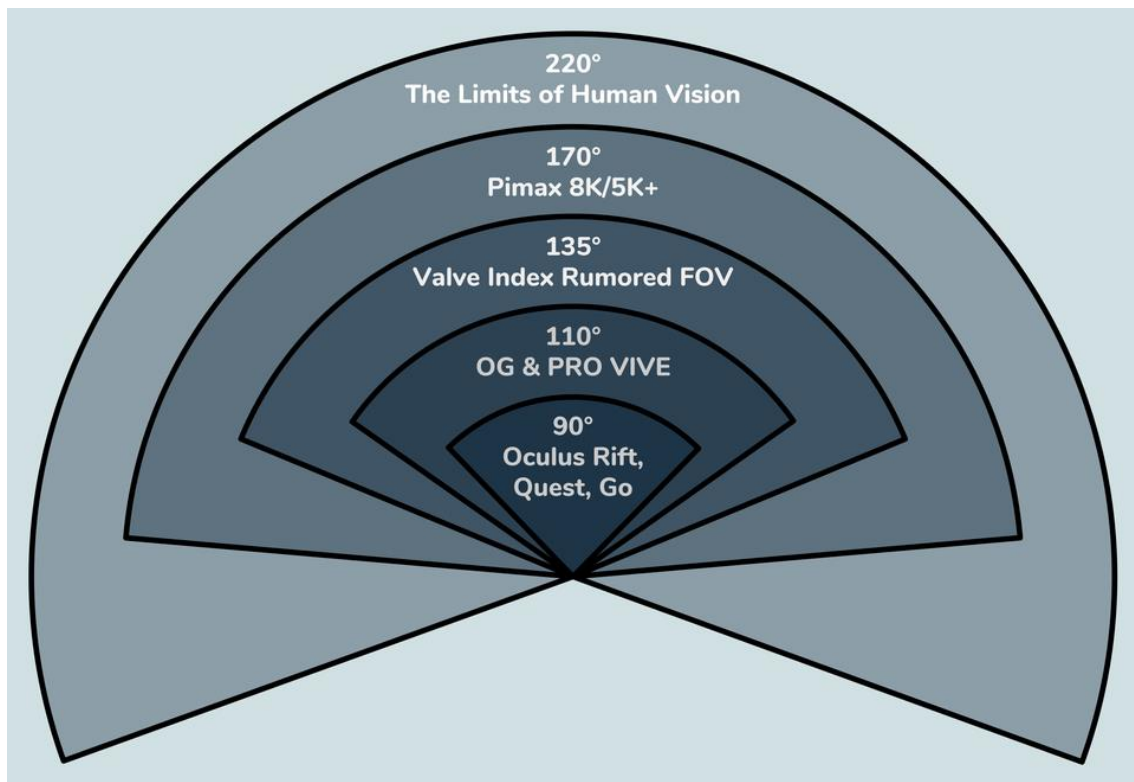


Figure 5: Different headsets and humans FOV

But maybe this is not the best idea to be honest. In Jason Jerald's *The VR Book*, he said "Humans have a strong bias toward worlds that are stationary. If some visual cues can be presented that are consistent with the vestibular system, motion sickness can be reduced" [8]. This already gives us some clues on why locomotion systems like Smooth Continuous might be the cause of so many cases of motion sickness.

As we mentioned, one of the theories that tries to explain motion sickness is the theory of Sensory Conflict, which happens when the brain receives two incongruous states of locomotion, in this case it would be what our eyes see and our vestibular system feels. With the smooth continuous locomotion we see ourselves moving in the game but we don't feel that movement, therefore causing motion sickness. The reduced FOV is the proposed solution for this problem, but it does not work exactly like it would work in regular Video games, here you cannot change the resolution any time you want or resize the image, no. The solution is to hide the surroundings.

As unbelievable as it might sound, the whole theory on how by hiding the surroundings you might reduce motion sickness comes from an optical illusion, based on Perspective Distortion [9]. Short focal length cause distances to expand, and longer focal lengths cause distances to compress. By "hiding" the peripheral vision we also change the perceived speed [46], therefore making the player feel they are not going as fast as they are and avoiding motion Sickness in the process.

Now, back to the remaining locomotion systems.

Arm Swinging/Headset moving: This is a new type of locomotion barely used in VR today. This method tracks either the headsets move or the hand controllers, displacing the player while you move them trying to emulate real running and walking. It is believed to have a less rate of motion sickness than the continuous smooth option, but it has one problem. If you are required to move your hands in order to displace from one point to another, can you use them for anything else at the same time?

Even though each one of these types may have variants, we shouldn't consider them as different types, mostly because these variants are fruit of the fight against motion sickness. Teleportation is a pretty simple concept, but the small change of adding a blink may change drastically how the players perceive this movement, becoming even less aggressive for them.

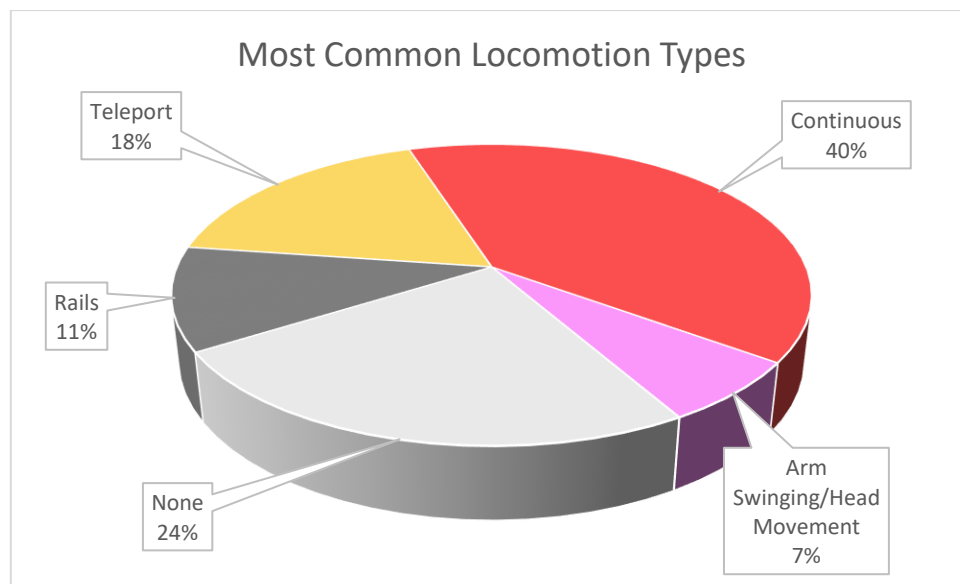


Figure 6: Most common VR locomotion types by percentage

Although some studies [10] like to make a distinction between Motion-Based locomotion and Controller-based locomotion to categorized VR locomotion I think it is a mistake. All movements are controller based, even those which use motion to move the player around, because all of them require an input from the player, they need to press one button or another to start processing the motion and transform it in locomotion. This difference is interesting not as a categorization method, it is interesting because it may be one of the keys to fight motion sickness.

Right now, VR locomotion is still in its very young, like VR technology in general, so the available options are few. On the other hand, these types of locomotion are not intrinsic of VR, each one of them were born from the necessities of the developers that created them. The arm swinging method for example has been around for only about three years. We could say that there is a long way ahead, but these few types of locomotion can pave the way to whole new types that right now we might not imagine.

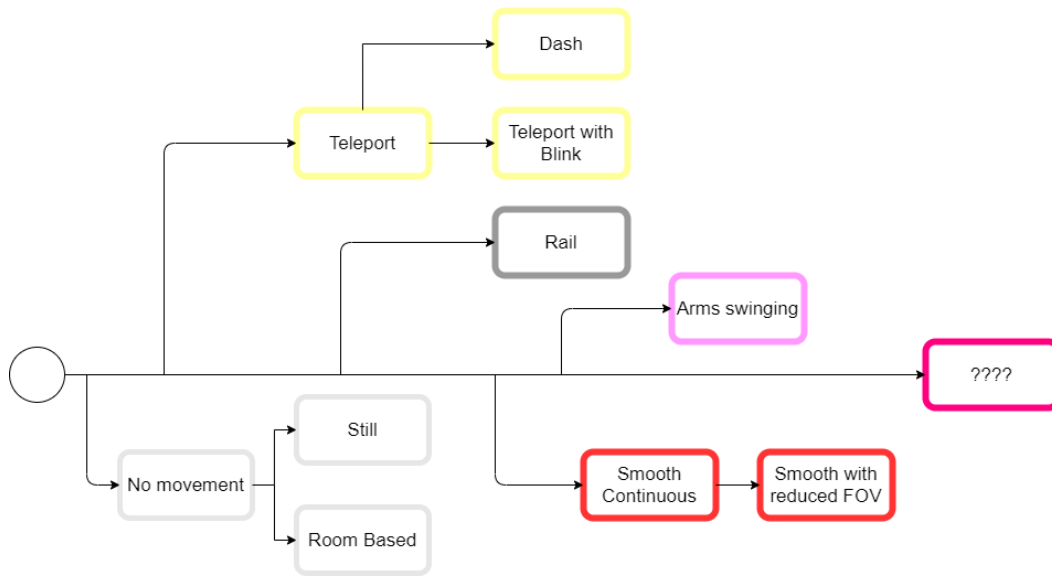


Figure 7: VR Locomotion "Family Tree"

From the information we have gathered we can even come up with some sort of "Family Tree" of the current locomotion systems that are used in VR. As I said before, each one of the five main types were born due to the needs of the developers, independently one from another, and from those initial types new versions and even different types of locomotion systems were born to satisfy the needs of both developers and players.

2.3. Personal Experience

I began mi journey in Virtual Reality when I started developing a VR video game which we called *HAT*[11]. My colleagues and I were not big fans of traditional VR locomotion systems, and wanted to experiment with the technology. As a result, we came up with our own locomotion system, flying. This system was implemented by attaching thrusters to both hands, each thruster applying a separated force in the direction the player's hands are facing, moving the player in the opposite direction letting them to move freely in three dimensions and fly somehow like the superhero *Iron Man*.

Along with the flying system, we also implemented an arm swinging system so the player could use both. When the game was advanced enough we tested a demo version in some events and had really good feedback, especially because almost none of the hundreds of people that played it reported any dizziness or nausea, only some vertigo product of the heights the players reached in the game.



Figure 8: Screenshot of the video game HAT

Here we formulated a new theory. It is known and proved that motion sickness comes when players see some movement that they are not actually doing, because they know how that type of movement should feel. So, what if the movement they are seeing has not been experienced by the body before? No human can fly, at least not with their hands, so in some way we are “deceiving” their brains, they do not know what to make out of it because this is new for them.

This theory could also explain why motion-based locomotion systems like arm-swinging and the use of motion platforms produce less motion sickness than other locomotion systems, because the player is actually moving, maybe not like they actually should, but close enough to make their brains believe it. This theory is not crazy enough taking into account that some developers try (and succeed) to reduce motion sickness by creating Zero Gravity experiences [12].

We also heard from some of our university teachers that one reason that might be influencing the game is the presence of static references, which we didn’t know we had. As we can see in Figure 8, players wore hats and jackets as part of the aesthetic of the game, both elements could be seen inside the game, being the hat always static because it followed the player’s head, and the jacket acting as a body reference, creating body awareness and making it feel more natural. This theory has been already proposed for example by Assistance Professor David M. Whittinghill,

whose research team at Purdue University's in the Department of Computer Graphics Technology, implemented a virtual nose inside their simulations, reducing motion sickness in their subjects [13]. For this reasons, one of our sub-variants in each locomotion system will consist in putting a static reference for the player, such as the hat and jacket.

3. Objectives

The original objectives of this essay were the following:

1. Analyse the most common control and movement methods in virtual reality environments.
2. Study existing systems and innovative proposals as possible study paths.
3. Design and implement a navigation system for use in arcades that allows moving through terrain and simulating flight.
4. Analyse with users the operation of traditional systems and the system devised in the project and compare the results of these tests.

This essay was going to be tested with the help of a colleague in a Virtual Reality Arcade, unfortunately, due to the Covid-19 pandemic, the arcade was no longer an option due to hygienic reasons [14].

In order to continue with the experimentation and testing, I acquired a VIVE Cosmos headset [15] and invited up to fifteen different person to my home, where I could, taking the proper precautions, let them do the testing with minimal risk for everybody's health.

Even though the testing won't be done in a VR Arcade environment, the results still apply, mostly because the games employed in this type of business are the same we can acquire at home.

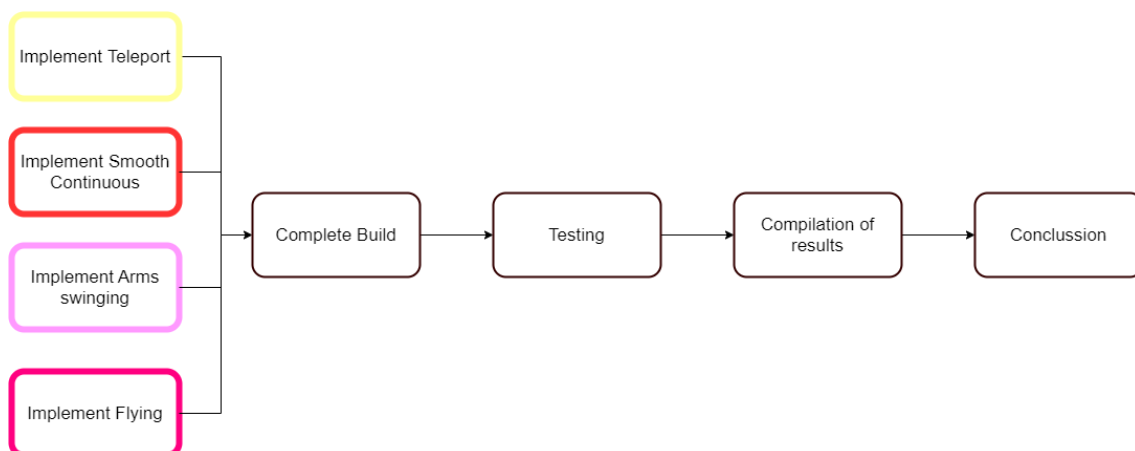


Figure 9: Project process

4. Methodology

4.1. Study Phases

This project is divided in two time periods, Pre-Study and During-Study.

The Pre-Study phase is everything this project is based upon, 2 years and 4 months of experimenting with VR locomotion during the development of the video game *HAT*. Of that period of time I cannot provide accurate documentation of which investigation process, development and experimentation was conducted. What I can say is that during this time I learned the fundamentals of programming and developing VR experiences using the game engine Unity [16], I programmed the different locomotion systems that will be used in this study and enough testing and iterations of those systems to perfect them.

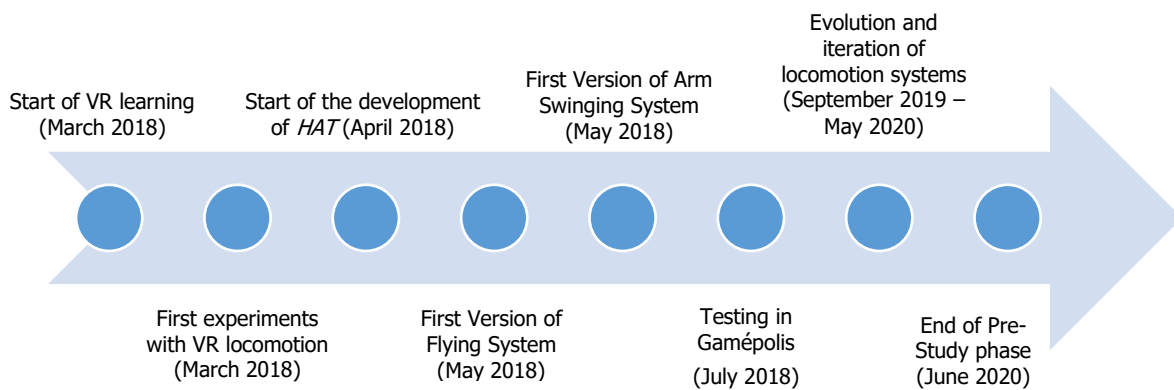


Figure 10: Pre-Study Phase Timeline

The During-Study phase is the work done since the start of this project on June 2020 to its delivery in September 11th 2020. This phase had a duration of 350 hours of work, divided in 24 hours each week. At the same time, the project was derived in four stages: Research, Development, Testing and Documentation. The amount of time spent in every stage was of: 40 hours of research and investigation, 120 hours building the demo (Development), 15 hours of testing and 175 doing the documentation.

Research phase: In the research phase I investigated what motion sickness is, what produces it, what solutions has been proposed, which types of locomotion are most common in VR videogames, came up with the testing questionnaire (Annex 1) and searched for other information that I might need during the development and documentation of the study.

Development phase: In the development phase I created the twelve prefabs that are going to be used by the testers, which includes hierarchy and each one of the different locomotion systems, already made before-hand but which needed to be adapted to this project. The design of the environment with animated characters, sound effects and lightning to improve immersion was also implemented in this phase.

Testing phase: The testing phase was carried in three sessions of five hours each. In each session five subjects participated. Each one of the subjects took around 30 to 45 minutes to complete its testing, leaving us the remaining time of the hour to clean the VR headset.

Documentation phase: In this phase I wrote this document. The document has been slowly written during the whole duration of the study while the searching and development phases were also conducted. When the testing was done, the final parts (Results and Conclusion) of this study were written, therefore leaving this document as finished and ready to present.

4.2. Working methodology

The working methodology employed in this study was *Get Things Done* or GTD [17]. GTD is actually a time management method, which consists in dividing our work in the different parts or tasks that conform it and start doing them one by one. When a new task appears, whatever the reason, is taken in to the pile of tasks and leave there until it is taken care of.

This method might seem lazy or inefficient, but that was not the case neither for me nor this project. This study has already a very natural established order: Understand what we are dealing with, design a testing environment, create that environment, test it with subjects and compile the results. Each one of these big tasks can be divided in smaller tasks, the creation of the environment for example can be divided in creating the scene, animating and placing characters, create the different player prefabs etc. Each one of those actions are independent, they will form a whole but they do not rely immediately in one another, so it's easy to just star doing work non-stop and then assemble everything.

This method is not recommendable when working in teams, using GTD was only possible because I had a global vision of how the whole project was going to be, so I subconsciously knew which order to follow to get the best results.

4.3. Project structure

The project will follow a very define hierarchy to keep it organized. This project will use assets from the internet and other personal projects to which we will add new elements which we will create in order to have this project working.

The project will have three main folders: Imported Assets, Downloaded Assets and Project. Along with this folders we will have several other folders from the *SteamVR* asset [18]. To avoid referencing problems we will leave them alone just in case.

Inside the Downloaded Assets we will place all the assets that we took from the Unity Asset Store [19], which include different models, materials, props and sounds.

Inside our Imported Assets folder we will store everything we took from older projects that will come handy, including materials and models but specially the original scripts and VR prefabs from which we will create the new ones for the study.

The project folder will store everything we actually use in the project, divided in several folders where we will store our own prefabs (Players, FOV...), the different NPCs with their animations and scripts to make them move and play their animations, the Animations we've got from *Mixamo* [20], and all the materials, shaders and scripts we used to create the scenes and improve immersion.

5. Economic study

In this point we will cover the economic cost of this entire essay, including software licenses and assets, hardware materials and the cost it would take to pay human resources for the development of this study in time and testing.

We used three paid assets from the Unity Asset Store, valued in the following prices:

- *Sci-Fi Sound Pack* by Cafofo [21] – 13.39€
- *POLYGON - Sci-Fi City Pack* by Synty Studios (Single Entity) [22] – 44.66€
- *POLYGON - City Pack* by Synty Studios (Single Entity) [23] – 17.97€
- *POLYGON - Heist Pack* by Synty Studios (Single Entity) [24] – 17.86€

Total: 93.88€

For the hardware we need a full VR ready PC as well as the headset, valued in the following prices:

- Computer HP Z1 Entry Tower G5 [25] – 800.88€
- Acer Full HD (1920 x 1080) IPS Ultra-Thin Zero monitor [26] – 75.94€
- VicTsing Wireless Mouse and Keyboard [27] – 19.40€
- Vive Cosmos Headset [28] – 799.00€

Total: 1788.28€

All hardware but the headset are not the ones that I used in the actual project because it was my own personal equipment the one it was used, and no tickets on the prices are available. For that reason, after some investigation, I found this hardware components which have the required technological specifications to run VR experiences.

Besides the testing done in the study with our subjects and the ones I did while scripting the different systems, early versions of the locomotion systems were tested in the 2018 *Gamépolis Video Game Fair*, in Málaga (Spain). In this fair a demo of our game *HAT* was open to the public for three days, with hundreds of people playing it and giving us feedback. In that game, as we mentioned in Section 2.3 of this project, the arm swinging and flying systems were implemented, so the feedback that we were given those days was very useful when we updated and fixed our programming.

The prices of the Gamépolis event are the following:

- Accomodation - 470.75 €
- Transport – 500€
- Diets – 150€
- Other – 100€

Total: 1220.25€

Some elements of the study have been in development since February 2018, long before the start of this study, but since they are part of it the time it took to finish them must be taken into account too. Therefore, we are talking about a complete length of 2 years and 7 months.

The development of this study can be divided in four stages and two periods of time. The stages would be Research, Development, Testing and Documentation and the time periods would be Pre-Study and During-Study

The Pre-Study Research and Development stages are by far the most extensive of all four. The mechanics and theories this study is based on started to being developed in February 2018, and were mostly done by a single game programmer. Working 15 hours a week, this developer has spent working in these two stages since February 2018 to May 2020 (included), a total of 1823 hours.

During-Study has been a work of around 24 hours a week since June 2020 to September 11th 2020, therefore rounding 350 hours of work. Of those 350 hours, 40 hours of research and investigation, 120 hours were spent building the demo (Development), 15 hours of testing and 175 doing the documentation, which also includes data compilation from the testing and all the interpretation of the results.

If we sum up both periods of time we get around 2173 hours of work. This might seem a lot and I'm 100% certain that we could reduce that amount of time in half. As I mentioned, this study was proposed after I had spent a lot of time and resources investigating VR on my own, because of the things I discovered and because I wanted to go even further. If it was the other way around, if the study was the original idea from which the whole experimentation came from, production times would be drastically cut.

According to *Indeed* [29], the average salary of a Software Developer in Spain 16.86€ per hour, to which we need to add a 30% due to Spain's Social Security cost, resulting in 21.92€. If we take the total amount of hours it took for the completion of this study (2173 hours) we get a cost of 47627.82€ total in a period of 2 years and 7 months to pay our Software Developer.

The resulting cost of this study through its two years and seven months of development is the following:

Concept	Price
Software	93,88 €
Hardware	1.788,28 €
Testing	1.220,25 €
Software Developers Salary	47.627,82 €
Total:	50.730,23 €

Table 2: Total cost of the study

As we predicted, the cost of this study is really high due to its origin, but I'm certain that if I all of this was started with the study instead that with a two years old game in development the costs of the Software Developer would drop by half at least. This is obviously because the focus would have been completely different and a lot of time that it was spent in different elements of the game wouldn't have.

6. Study Development

6.1. Summary

This study will consist in the implementation and testing of four types of VR locomotion systems in order to see, in comparison, how the players react to each one of them.

For that purpose, three of them will be common locomotion systems that we can find in different games on the market: Teleport, Smooth Continuous and Arm Swinging. The secondary purpose of this essay is to test a new locomotion system, the Flying system, so it will be implemented as the fourth time in order to see if it is actually viable as a new locomotion system for the industry.

As we already established, not only the system can be crucial, but also secondary elements such as the field of view [30] or having a static reference [13]. For that reason, each one of the systems will be implemented in three different ways: Raw, with reduced field of view and with static references. This way we will be able to see how this elements, usually implemented in videogames too, actually affects the player. This will make up to twelve different tests to carry.

For the testing I will use up to fifteen people. Each one of them will play each of the versions of locomotion and will answers a questionnaire which will be used to evaluate each locomotion system. Each locomotion system will be set in an individual scene.

The purpose and target for the subjects will be to find a key character in the scenario (Figure 11), a "Where's Waldo?" experience, which will change in each one of the different scenes. Once they find their target, they will be moved on to the next stage. It won't be actually necessary to find the target, it is just a measure to make them explore their environment, give them purpose and let them experiment for enough time with each locomotion system so they can form their opinion about it.



Figure 11: The target

6.2. Covid-19 measures

Having only one headset to carry the testing the danger of infection is really high, even if just one of the subjects is infected. For that reason a series of security measures were taken to ensure their health.

The most dangerous part are the controllers, being in contact with people's hands it could carry the virus and pass it around. Luckily this was quickly solved by asking each subject to wash their hands before the testing and also providing to each one of them a pair of latex gloves.

Following problem is sweat. VR headsets are in full contact with people's faces and it can reach high temperatures, that plus the movement the subjects will do means that they will sweat, wetting the headset and putting in danger the following subjects. The solutions for this problem were two: First we provided each subject a disposable mask and shower cap, to avoid the headset to touch most of the skin. Second, we establish the testing area in a room with air conditioning, to keep them as cold as possible.



Figure 12: Covid-19 security measures

Even though these two measures should be enough to avoid any danger, the headset and controllers were cleaned after each use with disinfectant.

6.3. Testing subjects and questionnaire.

For the testing we counted with fifteen people ranging from 22 to 29 years old. This is obviously not an ideal testing pool, neither in numbers or age range, but as we said, due to the Covid-19 condition, a small number of subjects was preferable to avoid unnecessary dangers.

Luckily the subjects come from different backgrounds and their experience using VR is really different from one another. The low age range is not ideal but not a huge problem either, according to the *GlobalWebIndex* study of virtual reality [31], done in the UK and US, around 35% of users between 16 and 34 years old have used a VR headset, more than any other age group, fading drastically in usage in older ages.

The subjects will require to answer a small and anonymous questionnaire. In this questionnaire, the subject will be asked to provide age and how many times have played any VR experiences before, choosing one of three options: Never, between 1 and 10 times or more than 10 times. This may be low numbers at first sight, but VR headsets are still an expensive product, so it is not very common to find people owning their own headset. At the same time, this question may

help us later to understand our results, after all, there is a probability that people that have played VR the most don't get as much motion sickness as the rest of users.

After this initial questions the subjects will engage in each one of the scenes prepared for them. After finishing their objective, they will be asked to answer in a scale from 1 to 10 how much "fun" the locomotion system felt for them and how much feeling of nausea and dizziness they got using it.

Due to the fact that we are dealing with the possibility of any of the subjects to have a big feeling of nausea we will provide them with the option of removing the headset and skip that locomotion system if they need it. In such case, the result in the dizziness scale will be of 10, but we still ask them how much they enjoyed it, just in case they want to give it a positive mark.

6.4. Unity and its tools

For the development of this study I will be using the game engine Unity. This is a very powerful tool and is the one I have been using the most in both academic and professional careers. Before diving in the actual development of each one of the tests it will be better to explain some key aspects of how Unity works and how we will use it.

Unity uses something called Scenes, the scenes are virtual environments, spaces where we will place all the elements that we will need in order to make our demo work, think of them as the different levels of a video game. In this study we will create twelve almost identical scenes, one for each type of locomotion system that we will implement.

The elements that will be placed inside the scene are called GameObjects, this elements will be everything that will conform the world, our city in this case: The buildings, the vehicles, the light, the characters... everything. GameObjects can be inside one another, constructing what is called as hierarchy. When inside a hierarchy, an object can be a parent and/or a child, depending on their position inside it.

Each GameObject will have different elements inside of them, which will give them different properties. Among those elements we should mention the most important for our study:

- **Transform:** This element keeps track and lets the user manipulate the position, rotation and scale of the GameObject. When inside a hierarchy, the transform properties will be relative of the parent. All GameObjects have a Transform.
- **Rigidbody:** This element gives physics properties to the GameObject, such as mass, air drag and even to be influenced by gravity.
- **Collider:** The colliders could be considered as the "skin" of the GameObject, they limit the bounds of it. It can have basic shapes such as spheres, cylinders and capsules, but they can also have the shape of a 3D mesh. Both Rigidbody and collider are needed for GameObjects to collide physically with one another.
- **Scripts:** Scripts are pieces of code which can affect almost every property of the GameObject and its elements. They are used to move GameObjects, create interactions between them and every behaviour and event that may happen in a video game.
- **NavMesh Agent:** This element allows a GameObject to use Unity's pathfinding algorithm to go from one point to another in our scene. To make it work, we must add a Navigation Mesh on the scene with the walkable areas that it will contain for our GameObject to displace on.

As we mentioned, scripts are used to control almost every aspect of our scene and GameObjects, this is done with the help of Unity Functions. These functions work as any other programming function that we might create, but they are part of Unity's engine and they tag specific aspects and attributes of the elements that form GameObjects. Functions like `Transform.Translate` and `Rigidbody.AddForce` for example will be used to move our GameObjects around the scene in the project, along with others which will have different functionalities.

The next things that we will use are Assets. Assets are downloadable packages that you can get from multiple places like the Unity Asset Store or other websites. These packages, which can be free or not, contain different elements inside depending on what they offer, they can contain 3D models, scripts, sound effects etc.

Plugins will be also needed for the project to work. Plugins are small subprograms that you can install on your Unity project to get more functionalities. In this project we will feature several of them, but the most important will be the SteamVR plugin for Unity, which will contain and install everything we need for our VR headset to work.

All these elements will be mentioned along this project, with their corresponding bibliographic references, please feel free to look them if any further information is required for their understanding.

6.5. Environment

In the early stages of the design, I thought of creating a unique environment for each one of the locomotion system. Later, I decided to use the same scenario in each one of the tests, because otherwise the test would not make sense. If each locomotion system have its own testing environment, there would be no way to compare them and get realistic results, all tests must be carried in a scenario where all of them can be used without problems.

One important part in VR and video games in general is immersion, to the point that, if the player does not believe the world he is in, it may even lead to motion sickness [32], so we need to create a rich enough environment to let the player feel the full experience.

I used two asset packages for this, the *POLYGON Sci-Fi City Pack* and the *POLYGON City Pack*, both made by Synth Studios and available in the Unity Asset Store for purchase. In the scene we will use the example City Model provided in the *Sci-Fi City Pack*. To that, we will add several more components to improve the immersion.

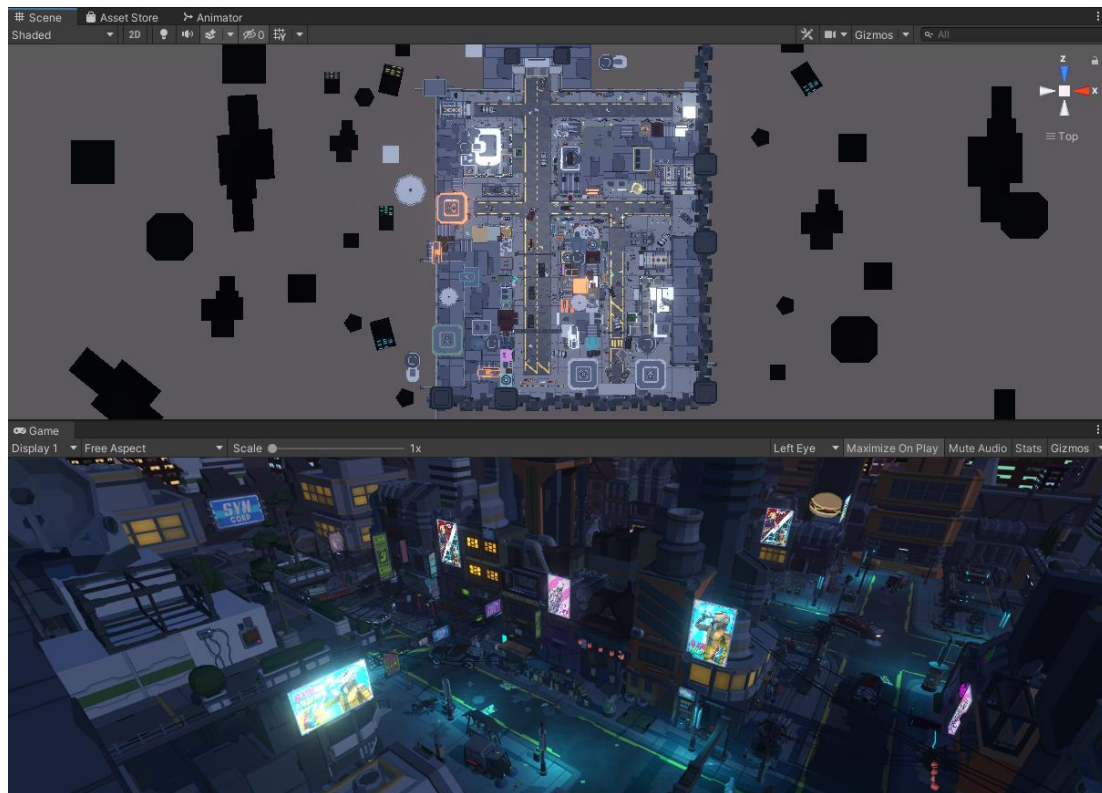


Figure 13: The testing scenario

The first thing we need to add here are characters. For this, we took the different models provided by the pack, and selected and downloaded a series of animations from the website *Mixamo.com*. The NPCs will populate the streets and buildings executing one set of animations or another depending on their places and which actions they are supposed to be doing.

To add more immersion, we also modified some of the models to carry weapons and other props, and even setting them in patrols or walking randomly around the streets. These NPCs will use a small script which will set as destination random positions in the Navigation Map [33] of Unity within a radius from its current position.

To avoid unnecessary problems, the NPCs will not have any type of collision with the player in order to avoid them to push the player when they are walking from one point to another.

The package also provided several flying vehicles and drones, so I implemented them as well. To get a smooth and customizable path I used the asset *Cinemachine* [34], provided by *Unity*. With this asset I defined several paths and carts through the city, each one of them with a different vehicle or drone.



Figure 14: Screenshots of the testing scenario populated and illuminated. Bank (Up) and Street (Down)

Next, it was the turn for the illumination. To achieve this Sci-fi look I updated the project to use the *Universal Render Pipeline* [35], which still has compatibility with VR. With this, I configured the pipeline and camera to make emissive materials glow and other effects to achieve the effect shown in Figure 14.

Finally, the last step was the audio. For that I downloaded several sci-fi sounds from the *Sci-Fi Sound Pack* made by *Cafofo*, also available for purchase in the *Unity Asset Store*. I used several of the sounds for the vehicles and drones, randomizing the pitch of each one of them to make them sound slightly different and using spatial blend to make the player hear them only when reaching certain distance to the player. At the same time, I added a city background noise, several effects in key areas like the Bank (Figure 14) and a music track in the Bar.

6.6. Device and key mapping

For the implementation and testing I used the headset *VIVE Cosmos*. For the implementation in Unity, we will need to download and install the *SteamVR* asset from the *Unity Asset Store*, this will give us access to all the libraries and scripts that we will need to detect and control both our headset and controllers.



Figure 15: Vive Cosmos

This headset has the particularity of not needing any tracking station. The headset has six camera sensors around it, which keeps track of the room and the controllers instead of relying it to external sensors, saving us time in room configuration and installation in each of the testing sessions.

SteamVR allows *Unity* developers to create their own input variables and map them in the controllers, this way each developer is able to have its own input and key settings. For our testing we will only need to use two of the buttons of the *Cosmos* controllers, the triggers and the joysticks.

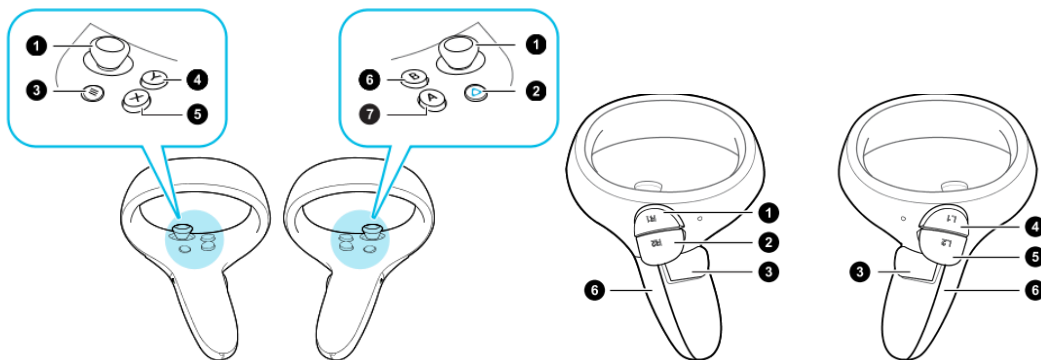


Figure 16: Vive Cosmos controllers – Joysticks (Nº 1, left side) – Triggers (Nº 5, right side)

For this buttons to work, we created two variables inside the *SteamVR Input* configuration in *Unity*, a *Vector2* variable called **Steering** which will be bound to the controller's joystick, and a *Boolean* variable called **GrabPinch** which will be bound to the triggers. With both variables and bindings created, we just need to reference them by their name in any of the locomotion scripts and we are good to go.

6.7. Locomotion Implementation

As we mentioned, four different locomotion systems, sub-divided each one of them in three different variants, will be tested. Each one of these variants will be placed in an individual scene, with the only difference being the prefab of the player and the location of the target. Each player prefab will count with the proper scripts and hierarchy needed for each system to work properly. Each prefab will be based on the prefab *Player* that we can download from the *SteamVR* asset from the *Unity Asset Store*.

6.7.1. Player Hierarchy

The hierarchy of the player will be based in the *Player* prefab provided by the *SteamVR* asset. There are three parts that we must keep at all cost, which will be vital for the player to actually work when using our VR headset.

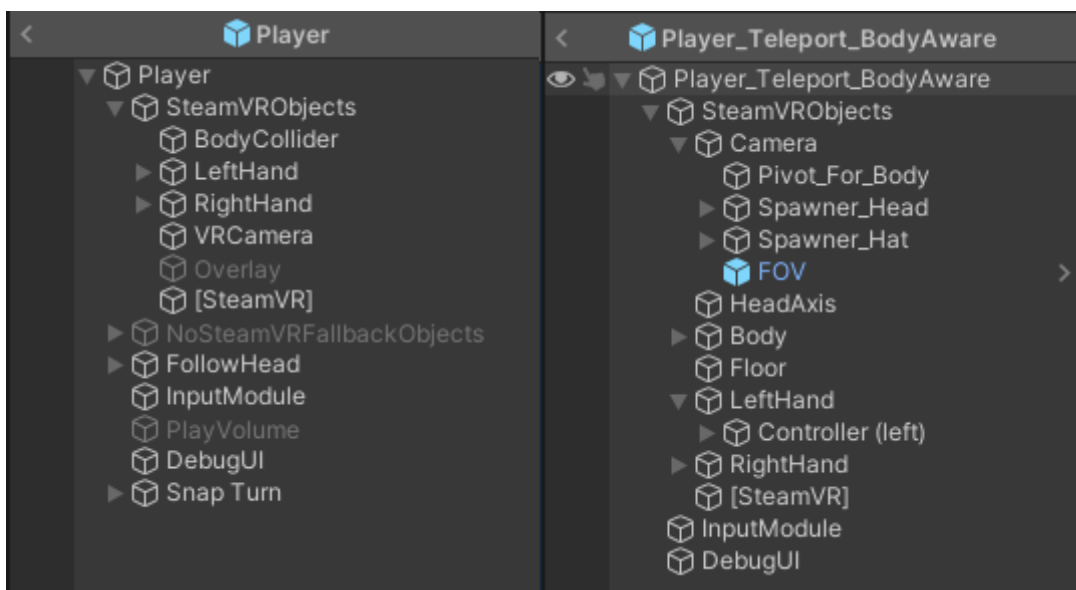


Figure 17: Original Player Hierarchy (Left) – Player Teleport with Static Reference (Right)

The first part we need to leave always and understand how it works is the top element of the prefabs hierarchy, the Player prefab. This one is not only the player but also the room area as well, all the tracked zone the player is allowed to move around in real life.

The second most important part is the *VRCamera*, this element is the headset device, which moves independently inside the Player prefab as well our heads moves inside our tracking area. We need to understand this difference between both prefabs because these two elements are the key component when we use locomotion. When using locomotion we need to move the whole prefab, tracking area included, independently where the head is placed or looking. The rotation of the camera will allow us to maybe determine the direction the player will follow, but we won't move it by code.

To have collisions and gravity we need to implement inside the Player prefab a Rigidbody Component [36] as well as Capsule collider [37]. The first thing you'll notice when doing this is that the capsule collider is not following the player because the headset is a child object, it moves independently and does not affect any components of its parent. To solve this, I came up with a script which does two things. First, it measures the difference in height from the headset to the prefab Floor in the hierarchy, and with that height and the position of the headset it will resize and reposition the centre of the capsule collider to follow the player anywhere inside the tracking area. That way we achieve body collisions wherever we go.

Last, and not least important, are the hands. The Player prefab have inside its hierarchy a couple prefabs called *RightHand* and *LeftHand*, each one of them with a script called *Hand* provided by the *SteamVR* asset. This script will be in charge of spawning the prefabs of our hands, fully animated. As well as the headset, the hands will move independently depending on where the controllers are in our tracking area. Inside of this prefabs we can put any prefab we need to improve immersion, for example a couple of Particle Systems [38] to simulate the thrusters in the Flying locomotion.

6.7.2. Raw, Reduced FOV and Static Reference

The three variants that each locomotion system will be derived in to are: Raw, Reduced FOV and Static Reference. The reduced FOV [30] and the presence of a static reference [13] are two methods suggested and implemented by VR developers to reduce motion sickness, but maybe their presence may affect the games playability and not be as good as a solution for all locomotion systems.

The Raw method will not have anything by itself, it will be just the prefab of the player with the pertinent locomotion system. Although, one change was implemented. The default method of the Teleportation system used by *SteamVR* already uses the blinking method by calling its own Fade-In Fade-Out mechanic. For both the Raw method and the Static Reference method this was needed to be removed, so I search inside the teleportation script where the Fade script was being called and put them under a Boolean condition.



Figure 18: Player using Raw variant

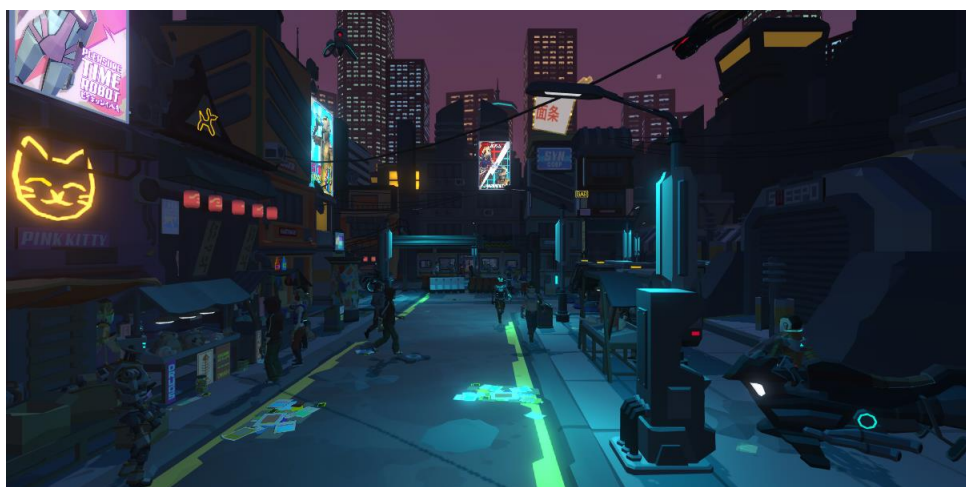


Figure 19: Players view using Raw variant

The Reduced FOV will be different depending on the locomotion system.

For teleport I will use a blink effect, that means that, when the player teleports, first we will turn the screen black, teleport the player, and then restore the view. What we will do for the remaining locomotion systems is to put in front of the camera a black plane with a hole in it and gradually modify the alpha level when the player is moving, that way the surrounding view will get blocked. Both systems will be activated from the locomotion scripts and will be placed in a prefab called FOV as a children of the camera (Figure 17), with a custom shader [39] that ignores the Z axis when rendering so it is always rendering in front of everything.



Figure 20: Player using Reduced FOV variant



Figure 21: Players view using Reduced FOV variant



The static reference is much simpler. Following the model we used in our original project *HATI* just put our old hat and jacket in to the players prefab, establishing them as our static references. The hat prefab will be a child of the camera, so it will follow it everywhere.

On the other hand, the body must follow the head but only in position and in the Y axis in rotation, after all, our body does not move when we tilt our head up or down. We want to be able to see our body if we look down, but if we rotate around ourselves we want the body to rotate with us.

A script was implemented to solve this. This script must follow the rotation of any given GameObject in the axis we desire, in this case it will follow the Y axis of the head, and this is achieved by simply getting and following the Euler Angles of the target in that axis. The GameObject with this script will work as a pivot for our body, which we can make as a child of the pivot or implement a secondary script if we want to have some thresholds for our rotations.



Figure 22: Player using Static Reference variant

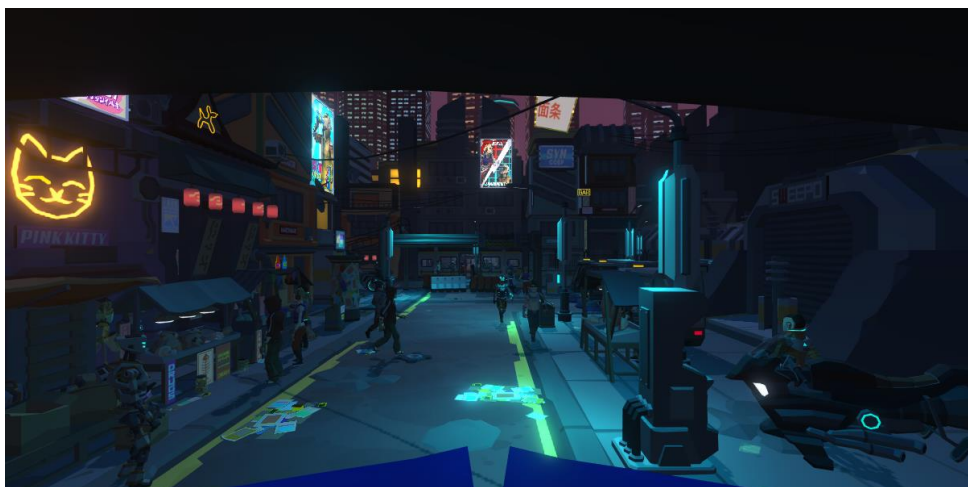


Figure 23: Players view using Static Reference variant



6.7.3. Teleport

Teleport is currently the most standard locomotion system you can find, it is the “default” system. Being this the case, the *SteamVR* asset already has a built in teleportation system which we will take advantage of.

In order to use it we need to keep the standard scripts that come with the Player prefab in the *SteamVR* asset, those scripts will already have configured the necessary inputs and dependencies to make the teleportation work. Beside those scripts, we need to use the Teleporting prefab that also comes with the asset. Inside this prefab, we will set which areas we want to be able for the player to teleport to.

This teleportation system works the following way: When pushing any of the joysticks a Bezier pointer will appear from the player’s hand, colliding with everything in the scenario. If the collision is against one of our Teleportation Areas, the pointer will turn green making it visible for the player that teleportation to that destination is possible, and when the joystick is released, the teleportation to that destination occurs. Otherwise, if the collision zone is not valid, the pointer will turn red and simply deactivate when releasing the joystick.



Figure 24: Valid teleportation point



Figure 25: Invalid teleportation point

To implement it in our scene we just needed to create our three different variants of the player using teleportation. Once placed inside the scene, we added the Teleporting Prefab, and inside of it we extended the Teleportation Area to cover most of the scenario so the player is able to teleport anywhere we wanted.

A couple problems came up when I implemented this method. When updating to the Universal Render Pipeline of Unity, all materials and several scripts were affected, among them the *SteamVR* materials of the Bezier pointer and the *Steam VR_Fade* script, which was the responsible to turn the screen black in the blink variant of the teleportation.

The solution for both, luckily, were pretty easy. I am not the first person to update to the Universal Render Pipeline, so I was not the first person to also encounter these two problems. A quick Google search landed me inside the *Valve Software Github* with the solution for the pointer problem, which was solved by just changing the shader for one I downloaded [40]. For the Fade-In Fade-Out problem I did not search for a solution. As I mentioned, I already needed to implement a black plane which alpha was modified for the Reduce FOV method, I could replicate the process but removing the hole by changing the texture of the plane. With this new black plane in front of the camera, the only thing I needed to do was to access the Teleport script, search the lines of code where the original *Steam VR_Fade* script was called and put my method instead.

6.7.4. Smooth Continuous

Smooth continuous locomotion seems to be the most used locomotion system right now (Figure 6), but somehow is the most controversial one in terms of inducing motion sickness. As I said before, this type of locomotion consists in moving the player continuously and smoothly (like walking) through the use of a set of buttons, joysticks or thumpads. The speed and the interaction with elements like stairs or steps may vary from one game to another, it depends entirely on the game and developer. This method is the most used in multiplayer games, especially first persons shooters, because other systems like teleportation may kill the game flow.

For our implementation we will use a script which will affect the whole prefab, the parent prefab, as we need to move the whole tracking area when we walk. This script will read the input from the joystick of our right controller, which will return a two dimensional vector. For its use we will need to transform it in to a three dimensional vector first, because we want to move in the X and Z axis, we need to keep the Y value of our movement at zero. After that, the movement will be achieved with the use of Unity's `Transform.Translate` [41] function, by using our new three dimensional vector, multiply by the `Time.deltaTime` [42] and our own speed variable.

In order to rotate, we will need to make the translation happen in the direction the player is looking, luckily the `Transform.Translate` function can accept any transform as the relative space from which to do the translation, so we will use our heads pivot for this. This way, if the player pushes the joystick forward, it will always walk in the direction it is looking.

6.7.5. Arm Swinging

Arm swinging as I mentioned before, is one of the newest locomotion systems there is in the industry right now. It is a motion-based system in which the player, by pressing one or several buttons, enter in "locomotion mode", and by swinging their arms around they start moving inside the game. This is a clear attempt to simulate the swinging of the arms when we walk or run, so it is very common that the quicker you move the controllers or the bigger the swinging the faster you move.

To implement this method you need to track the position of your controllers and find a way to detect and measure the movement they are doing. In my implementation I used a script which stores the Y value of the position of the each controller in a list of floats, and every X time it

checks the average value using absolute values, after all we want to know how much did it move, doesn't matter the direction. The resulting value is the "speed", which will be used in in the locomotion script inside the Player prefab.

For the actual movement of the prefab we had two options in our hands, we could do like with the Continuous Smooth locomotion and use a Transform.Translate function, but this function modifies the transform, it does not give any force to the players physics. For that reason, instead of that function, I used a Rigidbody.AddForce function [43] applied to the player's rigidbody, limiting the maximum speed it could reach. This way, the player will feel the inertia of the movement when they stop moving, giving a more realistic feel of running. As we said before, we need to activate this locomotion mode, otherwise you would move any time you move your hands, so this script will only run while the trigger buttons in both hands are pressed at the same time.

There was a small problem with this method, due to the nature of the environment there is several steps and other small obstacles around, and this locomotion method had difficulties to overcome it, so after investigating a little I found a tutorial to solve this very same exact problem which seems to fix everything [44].

6.7.6. Flying

I would like to say that my colleagues and I are the original thinkers of this locomotion system, specially due to how funny is and how it barely causes motion sickness to any type of VR player, but those are the same reasons why I am happy to see other developers around the world implementing this locomotion system and other variants in to their own games [45], proving that we are not the only ones experimenting with new locomotion systems in VR video games.

As I mentioned in the beginning of this essay, I programmed the first version of the Flying system when working in the game *HAT* with some of my colleagues of 4FreaksFiction, back in February 2018. This system was conceived to be the main locomotion system in a competitive multiplayer video game, and after searching a bit, we did not found any precedents of what we wanted to do, only some examples with no physics and zero gravity. Our idea was to really fly, to throw yourself out here, in any direction and have full control while at the same time battling and taking advantage of the physics of your body. The system therefore needs two key components, a body with physics and a thrusting system which you can control.

For the body I implemented the system I mentioned before, I added a Rigidbody with a weight of 75 kg to simulate more or less the weight of a human and a capsule collider, this last one will resize according to the distance between head and floor and reposition to be in the same position as the head.

For the thrusting we decided that the easiest way to control the movement would be a flying system inspired in the superhero *Iron Man*. The implantation therefore consisted in putting a couple empty GameObjects in each hand, pointing with the Y axis in the opposite direction of the palm.



Figure 26: Thruster position and rotation in each hand

With the body and the thrusters in position I created a script which is in charge of the actual flying. Each thruster will be activated when you press the corresponding trigger in each hand, executing a individual force of 700 Newtons, making up a total up 1400 Newtons if both thruster are activated, making it possible for the player to stay on the air with only one thruster activated. This force is directly applied to the rigidbody of the player in one direction or another depending on the orientation and status of each thruster. The global orientation that each activated thruster is pointing adds to a new orientation vector, this resulting vector will be the direction to which the force will be applied.

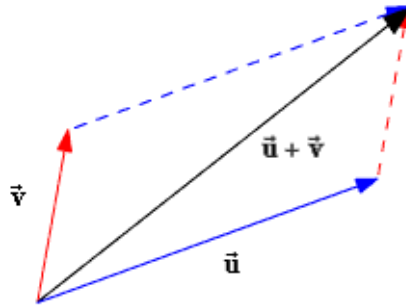


Figure 27: Addition of two vectors

This method has been evolving for two years now, iterating through several version which included a gas system, be able to rotate around yourself depending on the orientation of the thrusters (which was a bad idea, too erratic, too much motion sickness), different limit velocities, different air dragging... The version implemented in this study is the simplest yet the most efficient one, with no unnecessary mechanics that may affect the testing, just flying.

Unfortunately this system has a relatively big problem. As we are using real physics we need strong enough forces to move the player around, so it is inevitable to reach really high speeds, even by falling. By itself the system may cause vertigo in people who are not used to heights, we cannot deal with that problem because the cause has nothing to do with VR, but the landing and the crashing against objects has. Even with a limit speed established, the players will crash against buildings and fall to the ground at full speed, stopping in an instant. This sudden stop has been one of the main remaining reasons that can cause some nausea due to the sudden stop in movement. We are still working on some solutions for this problem that won't affect the playability of the system, but none of them will be implemented in this project.

For that reason, we will need to warn the players and make them think their responses when doing the questionnaire. When they answer how much nauseated they got while using this system they need to understand that we are asking for dizziness and nausea, not vertigo. Dizziness product of the sudden stops will be contemplated, but we must make sure they do not rise their dizziness marks because they are afraid of heights.

6.8. Development of the testing

The testing was carried in three different sessions of five people each through the course of three weeks. There was no critical problems in neither session.

The only problems I encountered were a couple bugs which were quickly resolved: A missing reference in the Teleport system with the Reduce FOV and a float which didn't have its correct value in the Smooth Continuous system.

Besides of that, a couple subjects reported that the room was too hot due to be carrying the testing on a summer and because of the Covid-19 measures, so I just lower the temperature of the air conditioner machine. No complaints about the temperature were reported after that.

7. Results

After compiling all the data from the fifteen test subjects, I proceeded to write it down in an Excel file, where I could represent the information in separate graphs in order to see clearly our results.

The full results can be found in the different tables in Annex 2. These results, as we introduced, will feature the subjects number, the number of times the subject played any VR experience in the past and the results it gave to both questions of dizziness and enjoyment in a scale of 1 to 10 in each one of the 12 tests.

Now, regarding this results. While doing the tested I already started noticing some kind of pattern in the numbers the different subjects were given to me, all depending on the number of times they played VR video games before, so I will highlight each "experience" group in the graphs with different colours: Blue for subjects that have never played before, orange for subjects that have played between one and ten times, and green for subjects that have played more than ten times.

In order to see the results even more clearly, I adapted the numerical results of the testing. With the original results we got a two dimensional graph that went from one to ten, therefore having the centre of the coordinate axis fixed in the bottom left corner of the graph. With this view the results were not as easy to read so I used a different representation method by "changing" the values.

First of all I added a small scatter so same value results don't overlap in the graphs. For whatever reason, subjects were giving really similar results in several of the tests, therefore results were constantly overlapping one another, making it look like some results were missing. With the scatter we can clearly see them now.

Second, I modified all results by subtracting five points of value to every number, that way the resulting graph is centred in both axis and we can define four separate panels which will represent the four possible types of results that we can get from the testing.

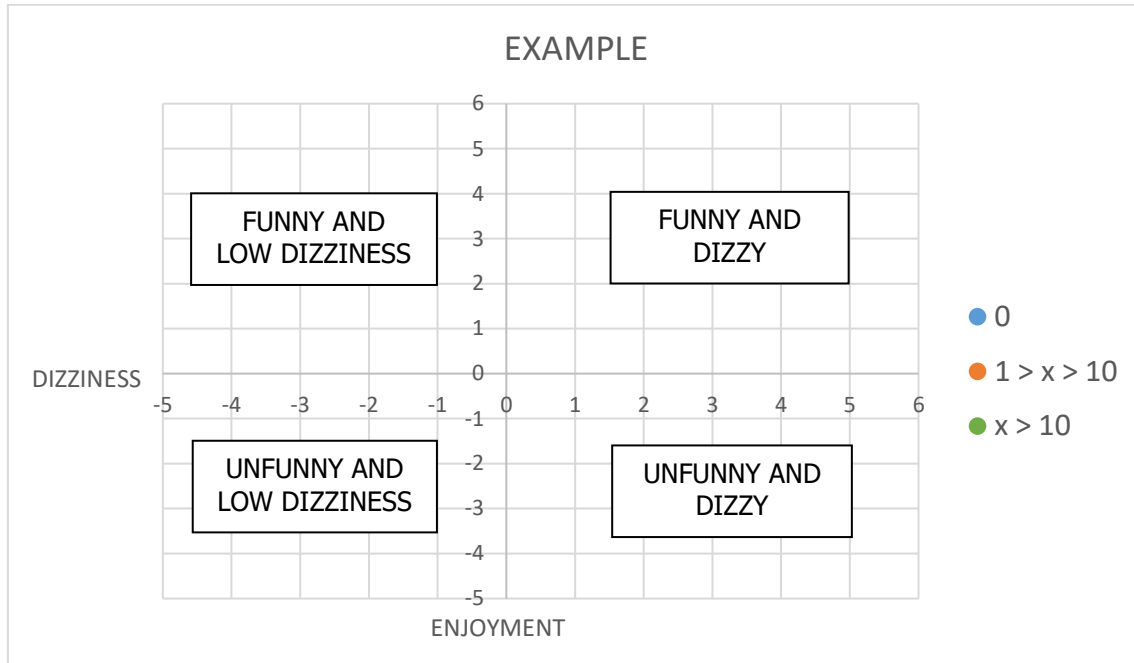


Figure 28: Graph example

The original values could oscillate between one and ten included, so in our graphs they will oscillate between minus four and five both included, giving an extra point on each side to give the graph some room.

Obviously this changes won't affect at all the results of the study, they have only been adapted for an easier reading. The original results, as we mentioned, can be check in Annex 2 of this project.

7.1. Teleportation

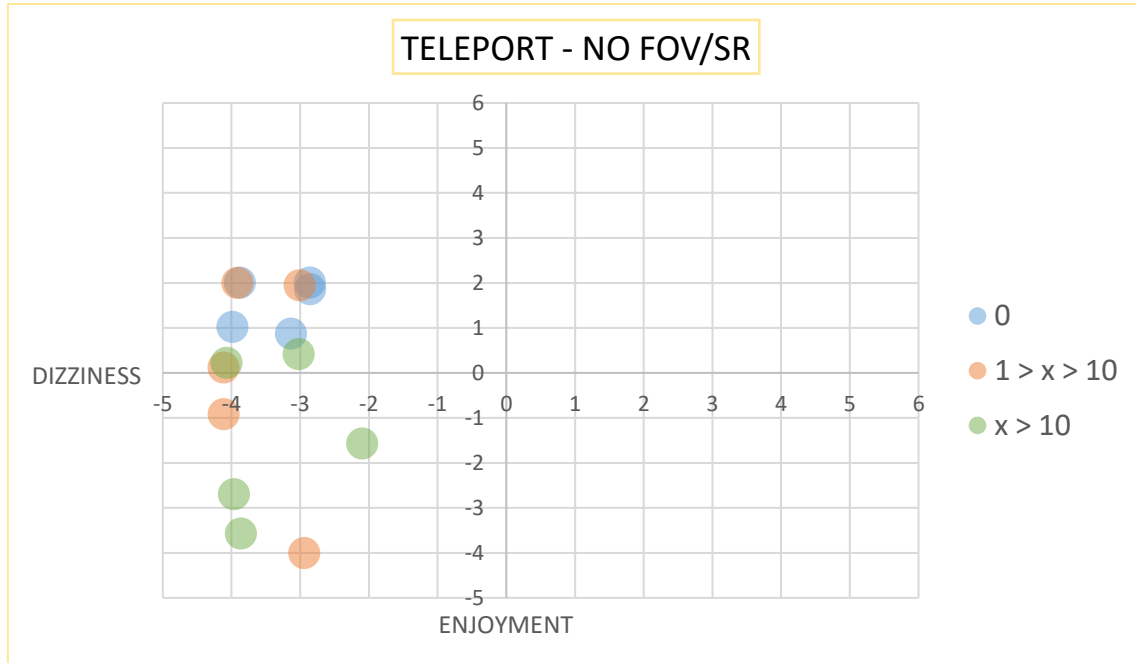


Figure 29: Teleport system graph without reduced FOV or Static Reference

In this first graph we can already see what I was referring to. The teleportation system without any reduced FOV (blinking in this case) or any static reference seems a fitting locomotion system when trying to find a system that avoids Motion sickness. All subjects seem to agree that the system by itself does not cause dizziness or only a little bit, but they differ when talking about how much they enjoyed using this system.

We can see that people that never played VR before (Blue) seem to enjoy the system, while people that have played more with VR seem to dislike it more and more (Orange and Green), possibly due to having experienced other locomotion systems which in comparison are funnier.

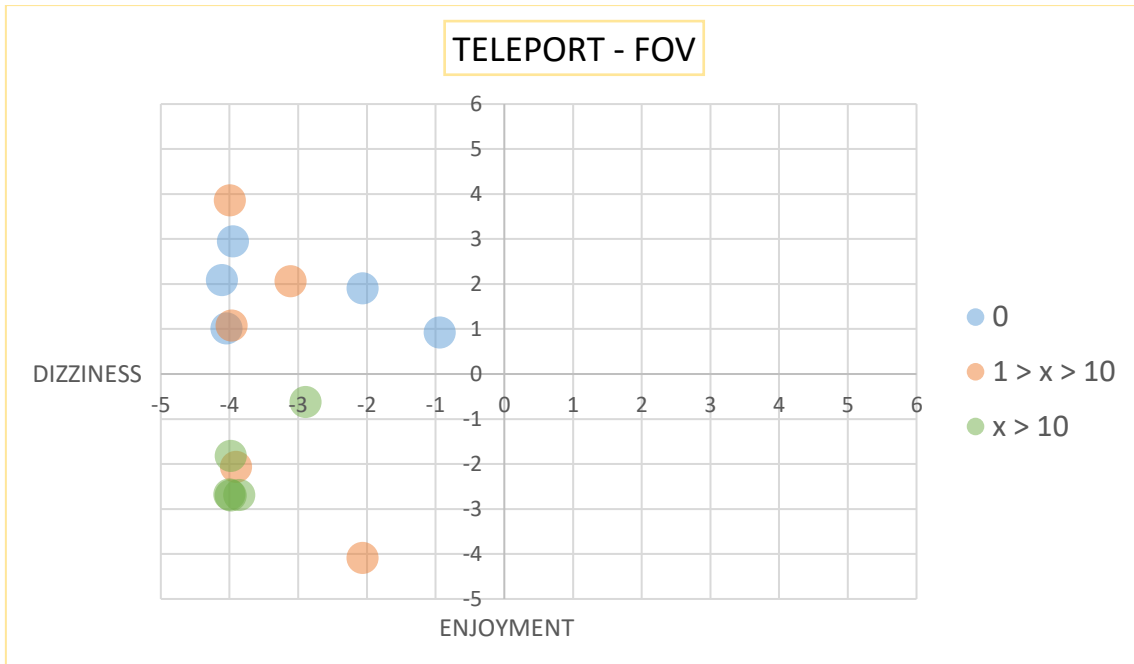


Figure 30: Teleport system graph with reduced FOV

As we mentioned, the Reduced FOV (or blinking in the case of teleporting) is used to reduce Motion sickness in theory. As we can see on Figure 30, the results tell otherwise.

While the fun have by the testers seem to stay more or less the same, the dizziness levels have increase in some cases, especially in players with less experience (Blue and Orange). I should note here that some subjects, even after explaining them what to expect, didn't notice the blinking effect at first sight, maybe thanks to the already intrinsic nature of the act of blinking.

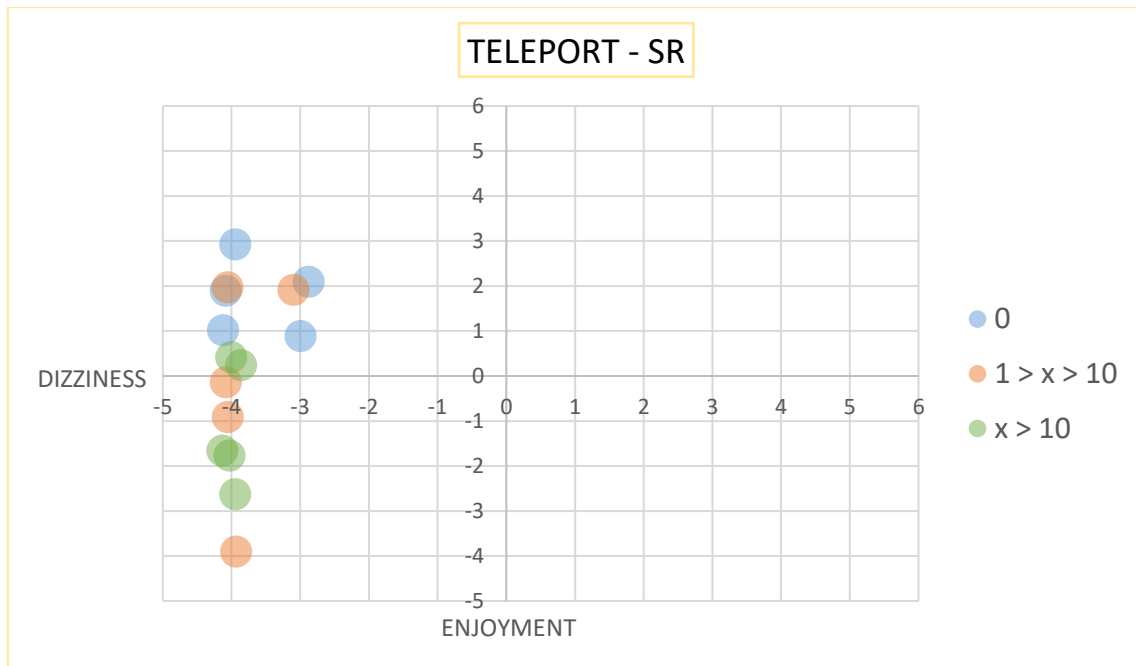


Figure 31: Teleport system graph with Static Reference

When using the static reference variant we see no dramatic change when compared with the first version without reduced FOV or static references. Note here that several subjects reacted positively to the fact that they had clothes and a body now, but it was not a major change and it did not make the locomotion system more enjoyable.

7.2. Smooth Continuous

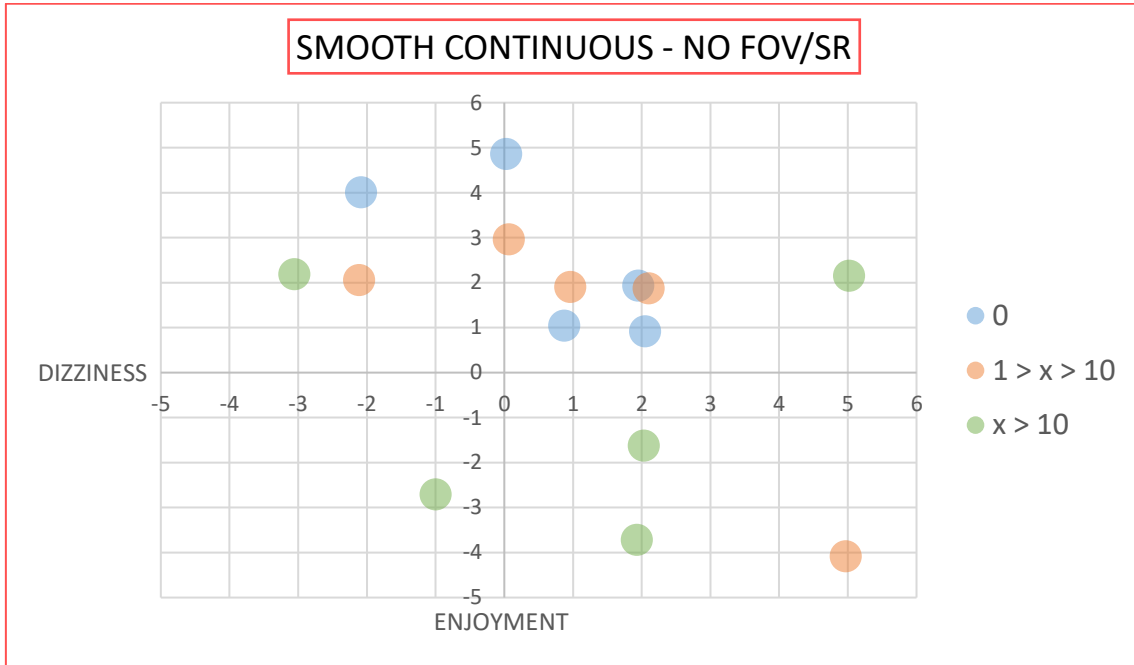


Figure 32: Smooth Continuous system graph without reduced FOV or Static Reference

The smooth continuous system, even being one of the most used right now in the industry (Figure 6), it is clearly from Figure 32 that it is not the one with the best response when fighting motion sickness.

From this graph we can see that it doesn't matter how much you have played VR before, the amount of dizziness and fun you experience is completely up to the subject's natural predisposition to suffer from motion sickness as if it were on a car or a boat.

Even so, it seems that people enjoy this locomotion system far more than the teleportation system.

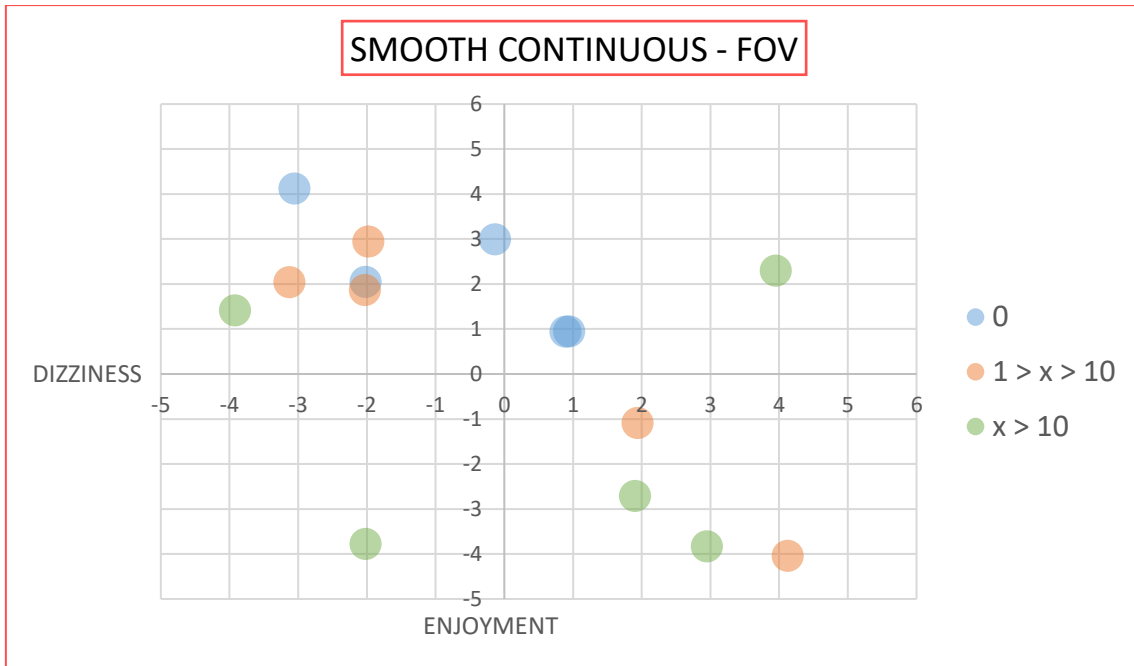


Figure 33: Smooth Continuous system graph with reduced FOV

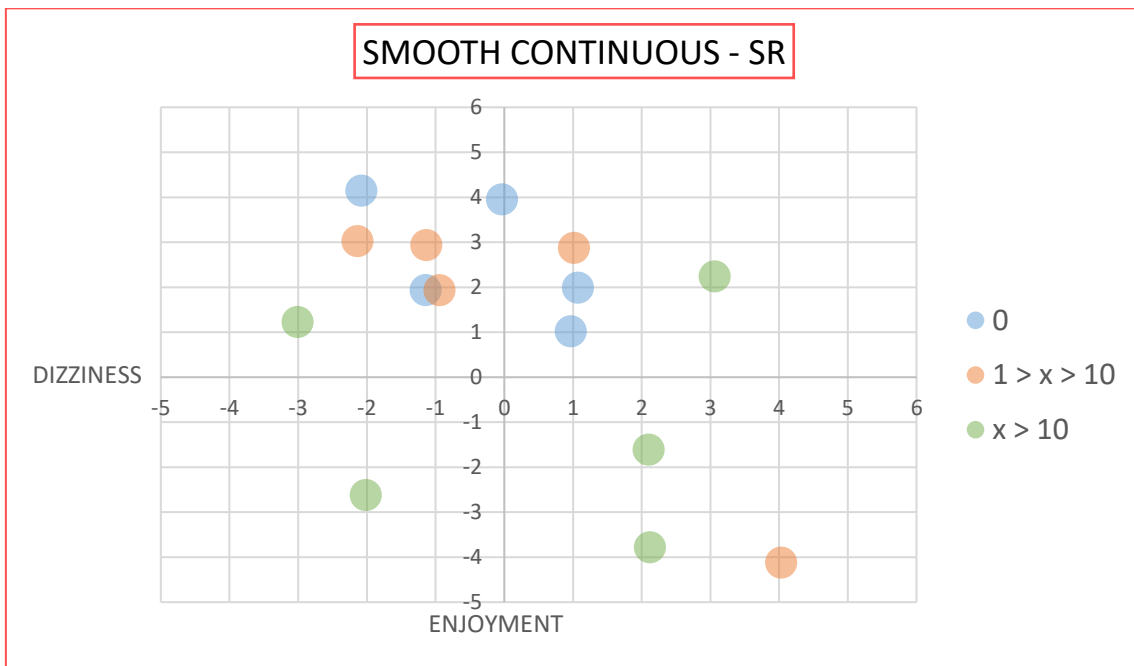


Figure 34: Smooth Continuous system graph with Static Reference

When applying reduced FOV or Static References the results remain chaotic and unpredictable. Although we can see some improvement in terms of Motion sickness.

7.3. Arm Swinging

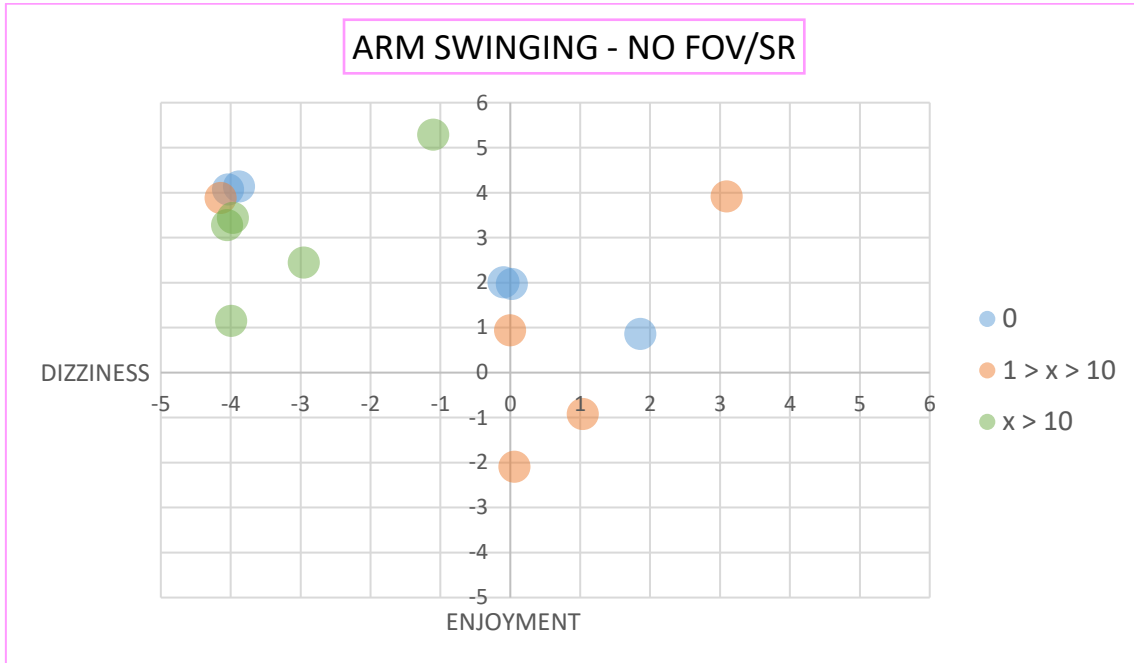


Figure 35: Arm Swinging system graph without reduced FOV or Static Reference

Arm Swinging responses seems to be clearly divided by the groups we established.

Both new players (Blue) and the ones that only have played a few times (Orange) seem to get more Motion sickness than experienced players (Green), and therefore they seem to enjoy it less, but they enjoy it anyhow. Experienced players (Green) seem to enjoy this locomotion system and barely feel any Motion sickness at all.

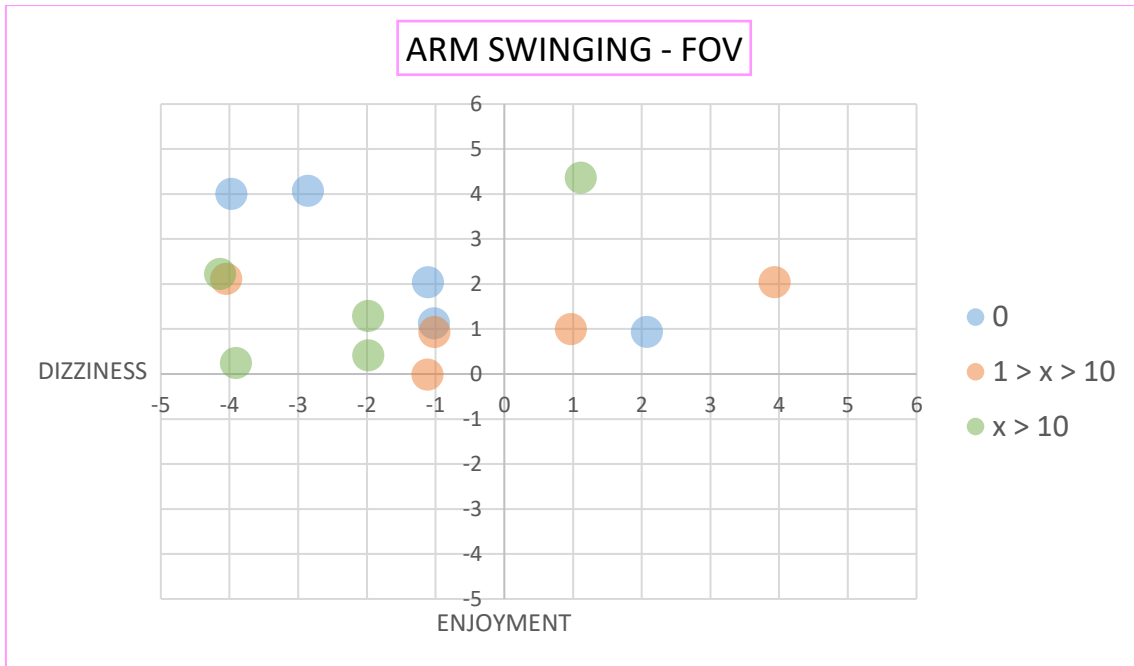


Figure 36: Arm Swinging system graph with reduced FOV

When implemented a reduced FOV we see clear changes.

All (Blue and Orange) players, except the more experience ones (Green), seem to have less dizziness sensation thanks to this method, raising their enjoyment level in some cases because they are able to play it better.

On the other side, experience player (Green) seem to rise their levels of Motion sickness in some cases and lower they enjoyment levels. When asked, they explained it was "weird" for them to not be able to see everything around them, lowering the immersion.

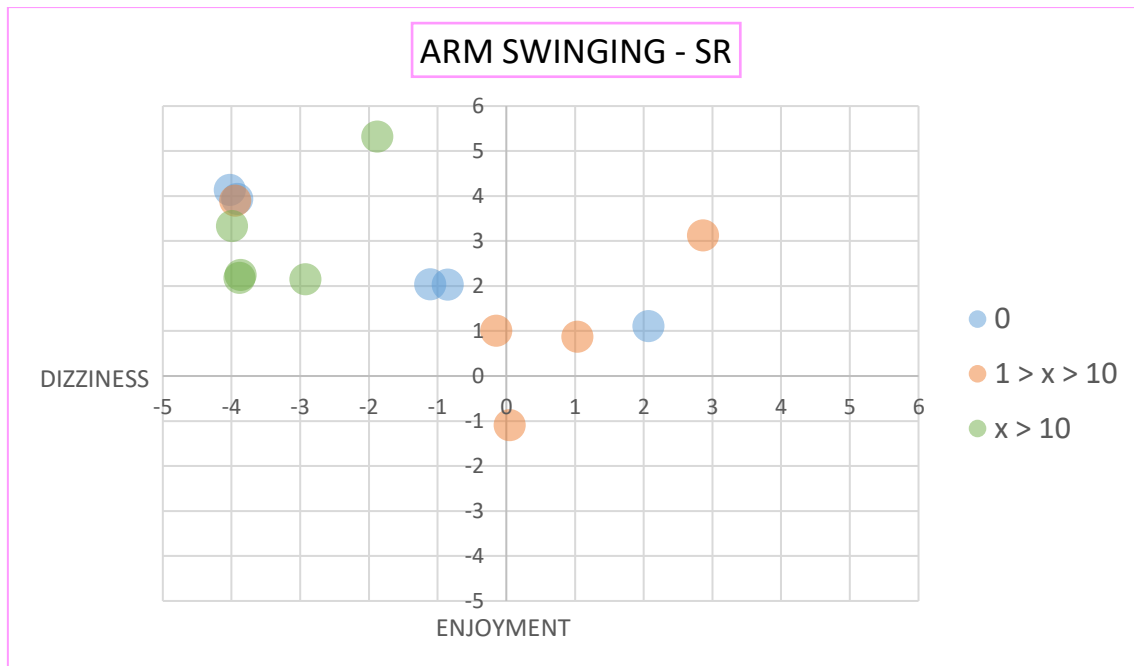


Figure 37: Arm Swinging system graph with Static Reference

When using Static References we get similar results that when we use nothing at all, but some cases seem to improve their levels of Motion sickness and a general rise in enjoyment levels.

7.4. Flying

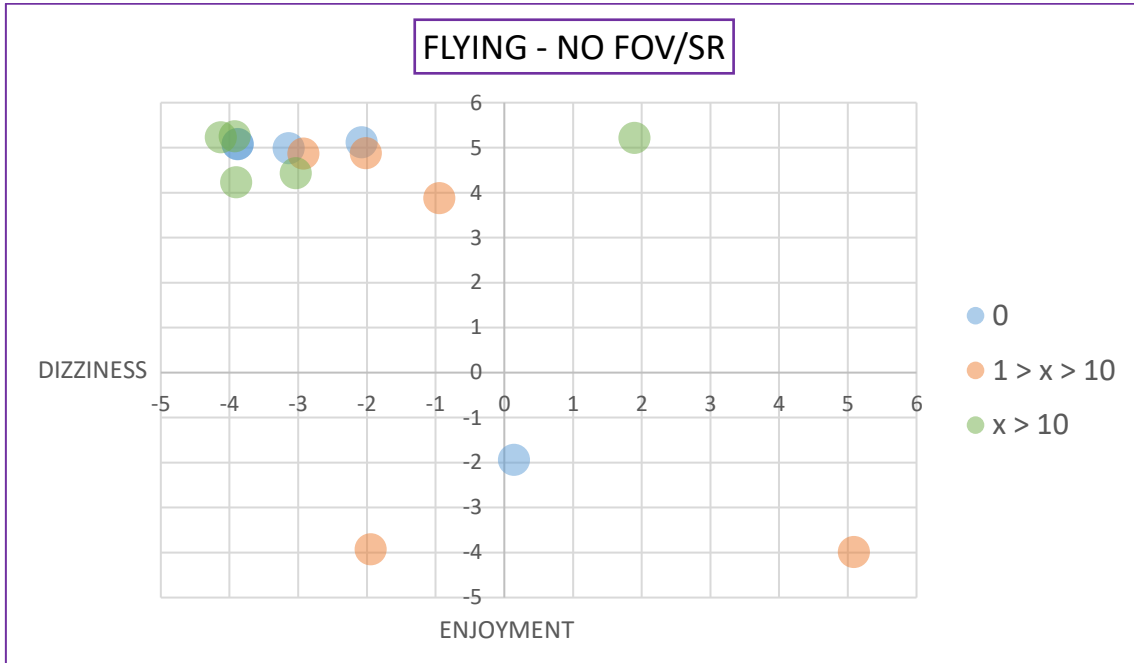


Figure 38: Flying system graph without reduced FOV or Static Reference

Flying seems to have one main response. For almost every subject, no matter their previous experience, seem to have a low rate of motion sickness and a high level of enjoyment.

It is noticeable that almost all fifteen subjects form a cluster in the upper left side of the graph (No motion sickness and funny), but four subjects are dispersed. If we take a look at the results table (Annex II) we can see that one subject did not enjoy the locomotion system due to the high level of motion sickness that the subject experienced. A couple of subjects did not enjoy the system even if they did not get a lot of motion sickness, when asked they responded that the sensation of vertigo was too much for them to handle. The last subject, a more experience one, seems to be the second subject to get the dizziest, but he enjoyed the experience completely, surprisingly at it may seem.

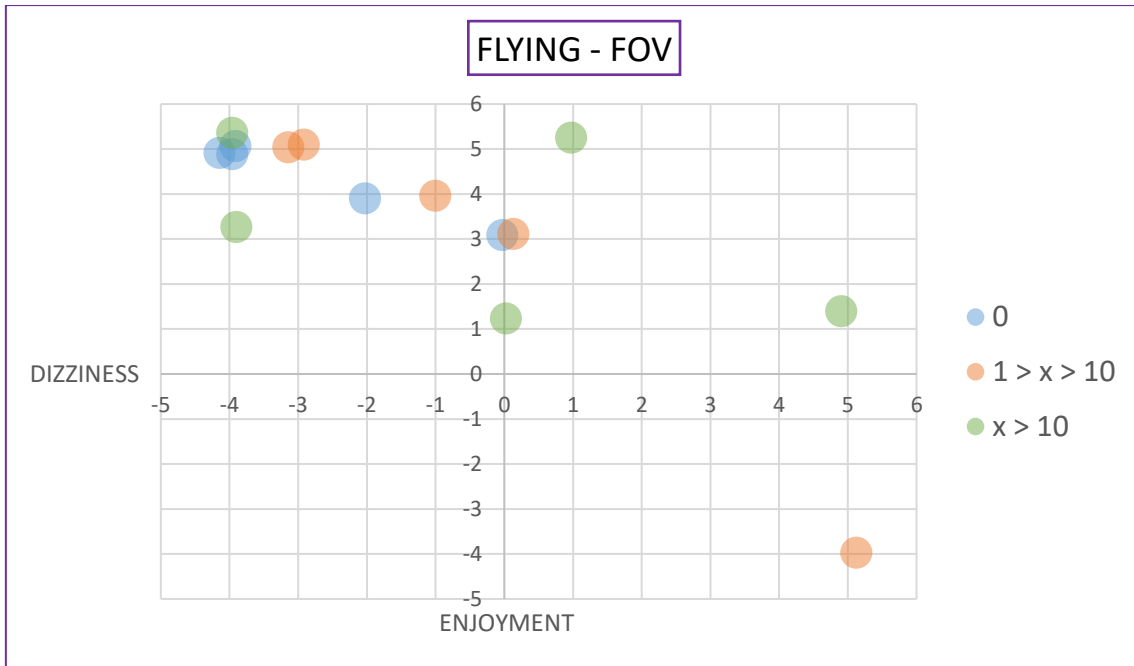


Figure 39: Flying system graph with reduced FOV

When applying a reduced FOV our results scramble. Even though some subjects do not seem to be affected, almost half of them see to get more Motion sickness than before, and do not enjoy the experience in the same level.

When asked, subjects explained that the fast movement was somehow crippled by reducing their field of view, making movement harder to control and to enjoy.

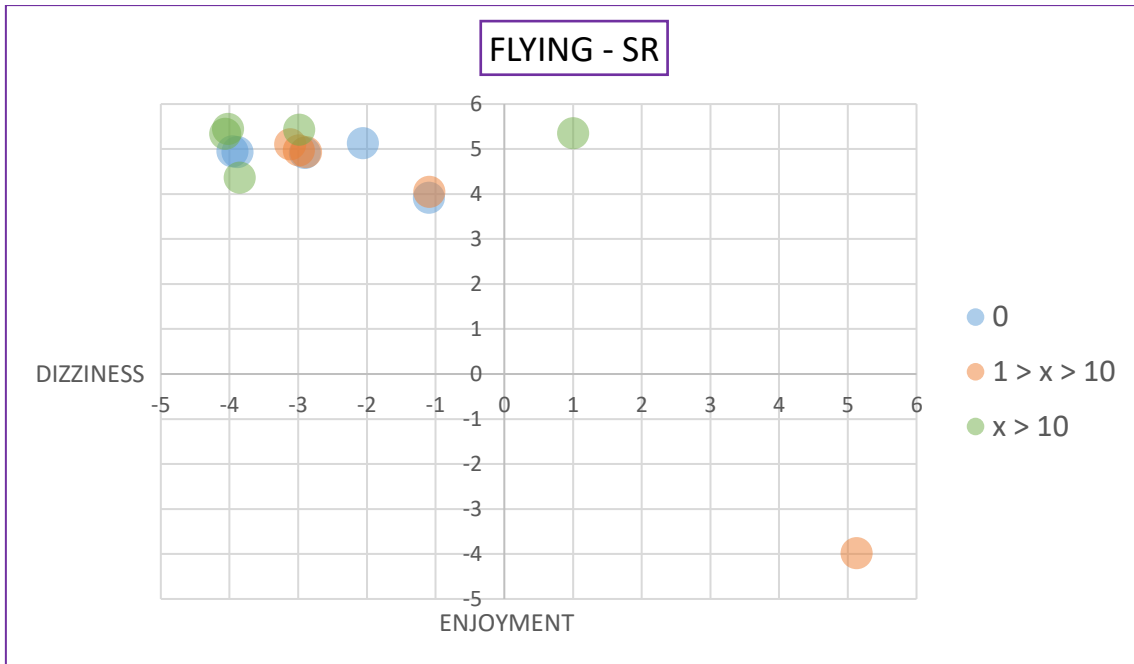


Figure 40: Flying system graph with Static Reference

When using Static References, we get even better results than when subjects do not wear anything at all.

Of the former four subjects that were "outside" regular levels, the two subjects that claimed to have vertigo changed their mind. They claimed that the references help them get less dizzy and therefore enjoy it more, but the main reason they dramatically change their opinion on how much fun the system is they claimed that it was just because they got used to the system.

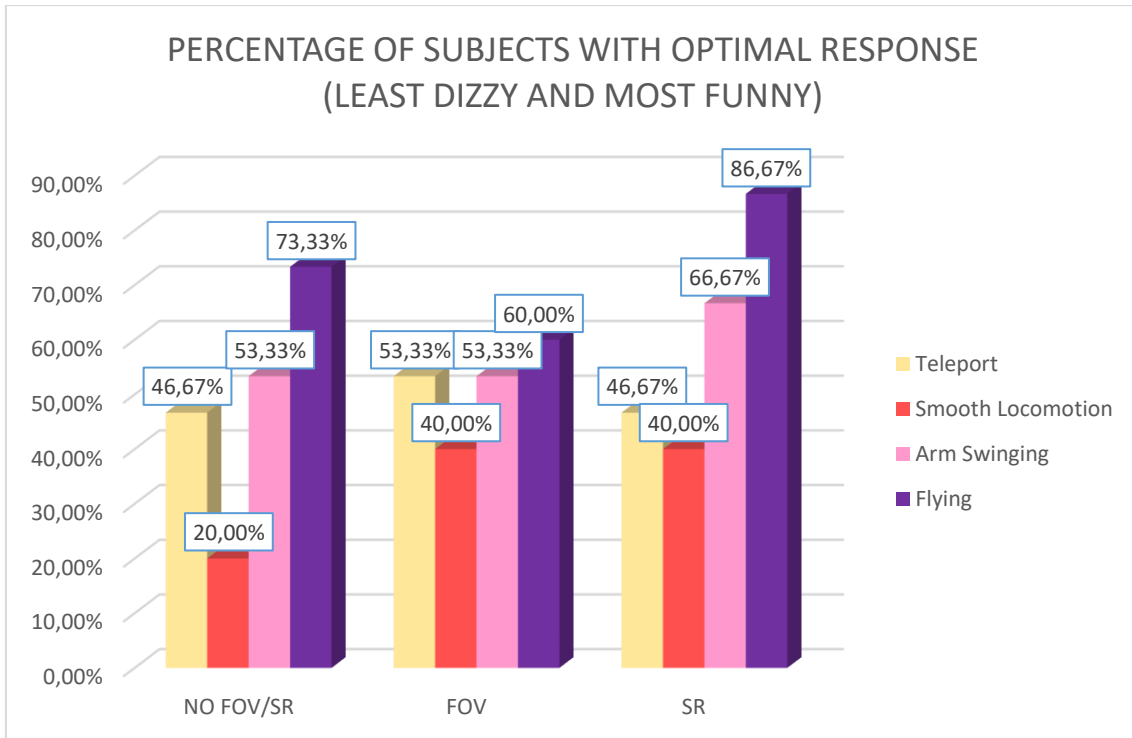


Figure 41: Percentage of subjects with optimal response

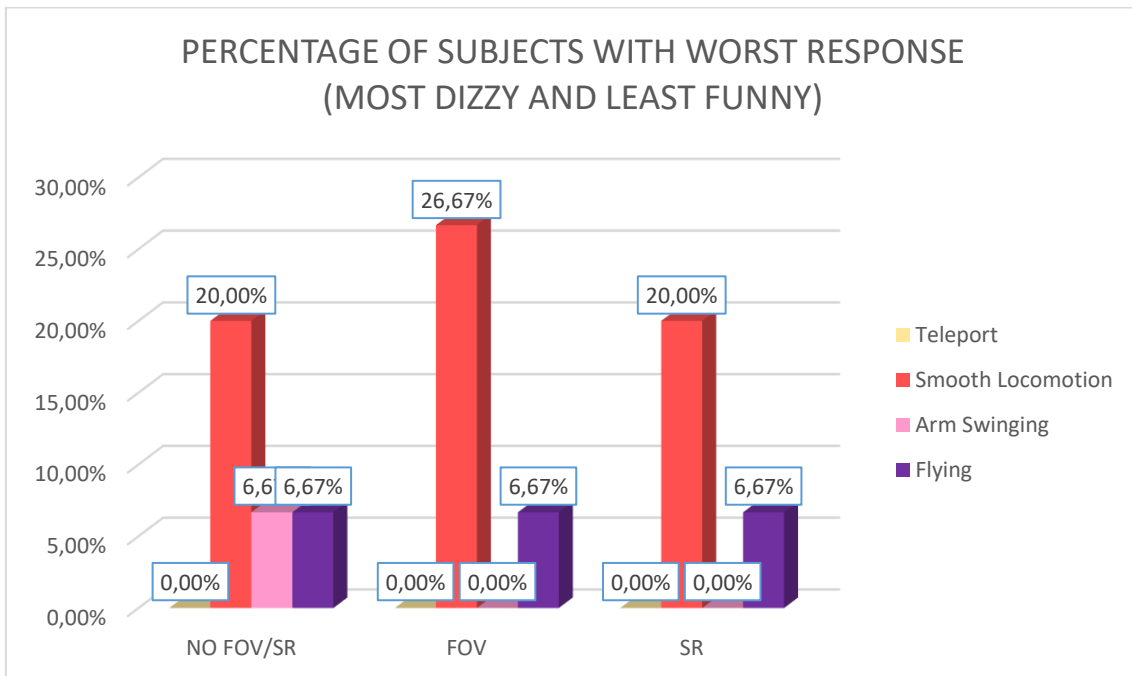


Figure 42: Percentage of subjects with worst response

When we compare all results by just looking at the most positive results and the worst results, we can clearly see a winner and a loser, but only depending on which criteria we use.

We can all agree that the Teleportation system seems to be the safest route to go, it convinces around 50% of the subjects no matter the variant and it has literally zero results in the worst case scenario when a locomotion system is both dizzy and not funny. From this we prove that it is not a surprise that is the default system to go when wanting to implement locomotion in VR as long as it does not affect playability.

Smooth Continuous goes the other way around. As I've been saying through all this study, smooth continuous locomotion is by far the least favourite of all four locomotion systems tested because is the one to that causes motion sickness most frequently. Even with popular methods like the reduced FOV or the static references we get really opposite results.

Arm Swinging, on the other hand, seems to be a really good choice, as we can see from the results in Figures 41 and 42, more than 50% of the subjects enjoyed the system and barely got any motion sickness, especially when adding the reduced FOV and the static reference.

But by far, the "winner" is the flying system, only 6.67% of the subjects (which is only one of the fifteen subjects) claimed the system to be unbearable, while the positives responses for it don't go below 60%, reaching up to 86.67% of positives response when using the flying system with static reference, proving that our theory, the "deceiving" of the brain, might be truer that we thought in the first place.

8. Conclusions

With all the results in our hands we can agree that we have learned several things with this experiment.

Locomotion systems are still in their early stages and they are very volatile. Even though we seem to start to understand motion sickness and methods to avoid it, we have proof that solutions like the Reduced FOV or the use of Static Reference do not always make a big difference or any different at all. We cannot predict how every player will react to our locomotion systems, their physiology and their experience will be the determine factors.

The player`s experience is, in my opinion after seeing the results, the actual key for the future of VR and locomotion in general. We have seen a direct relationship between experience and adaptability with more frenetic and fast locomotion systems. This experience, this practice seems to allow players to separate mind from body and be more adaptive to new movements.

At the same time, the mind is not the solely responsible on how players experience new locomotion systems. I have seen, not only in this study, but in other experiences and even in myself, that players, the so called *Gamers*, are always looking for new experiences, they want to go further, they want to explore new limits. I think this is where the key for the surviving and evolution of VR will reside, by constantly innovating. They will "force" themselves to learn and adapt.

The flying locomotion systems is the perfect example. It was born from people that were already used to conventional VR experiences and wanted something else, they wanted to feel things that no other video game platform could actually provide. It also proves that motion sickness, when understood, it can fought, proving that it is possible to create experiences that no only experience players can enjoy, but new ones too. From our study we can come to the conclusion even that the Flying is the best locomotion system right now due the high percentage of positive response.

In conclusion, I`m not the first person to write about this subject, neither will I be the last. VR will keep growing, new experiments will be carried, new experiences will be created and a lot more people will get sick in the process, but that`s the fun of it.

Annex I – Questionnaire

VR LOCOMOTION TEST QUESTIONNAIRE

Subject N°:

Age:

How many times have played any VR games?

- None
- Between 1 and 10 times
- More than 10 times

Teleport (No FOV – No Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Teleport (With FOV)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Teleport (With Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Smooth Continuous (No FOV – No Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Smooth Continuous (With FOV)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Smooth Continuous (With Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Arm Swinging (No FOV – No Static Reference)

Time:

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Arm Swinging (With FOV)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Arm Swinging (With Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Flying (No FOV – No Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Flying (With FOV)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Flying (With Static Reference)

Dizziness (1 None, 10 Unbearable)

1 2 3 4 5 6 7 8 9 10

Enjoyment (1 None, 10 Most)

1 2 3 4 5 6 7 8 9 10

Notes:

Annex II – Tests Results:

SUBJECT	AGE	FREQUENCY	TELEPORT - NO FOV/SR		TELEPORT - FOV		TELEPORT - SR	
			DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT
1	24	x > 10	1	5	1	2	1	5
2	23	x > 10	2	5	2	4	1	5
3	24	1 > x > 10	2	1	3	1	1	1
4	25	1 > x > 10	1	7	1	9	1	7
5	23	x > 10	1	1	1	3	1	3
6	23	x > 10	3	3	1	2	1	3
7	23	0	2	7	1	8	1	8
8	27	1 > x > 10	1	4	1	3	1	4
9	24	1 > x > 10	2	7	2	7	2	7
10	23	0	2	7	3	7	2	7
11	22	0	1	7	1	7	1	7
12	29	1 > x > 10	1	5	1	6	1	5
13	24	0	1	6	1	6	1	6
14	22	0	2	6	4	6	2	6
15	27	x > 10	1	2	1	2	1	2

SUBJECT	AGE	FREQUENCY	SMOOTH LOCOMOTION - NO FOV/SR		SMOOTH LOCOMOTION - FOV		SMOOTH LOCOMOTION - SR	
			DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT
1	24	x > 10	7	1	8	1	7	1
2	23	x > 10	7	3	7	2	7	3
3	24	1 > x > 10	10	1	9	1	9	1
4	25	1 > x > 10	5	8	3	8	4	8
5	23	x > 10	2	7	1	6	2	6
6	23	x > 10	10	7	9	7	8	7
7	23	0	3	9	2	9	3	9
8	27	1 > x > 10	3	7	2	7	3	8
9	24	1 > x > 10	6	7	7	4	6	8
10	23	0	6	6	3	7	4	7
11	22	0	5	10	5	8	5	9
12	29	1 > x > 10	7	7	3	7	4	7
13	24	0	7	6	6	6	6	6
14	22	0	7	7	6	6	6	7
15	27	x > 10	4	2	3	1	3	2

SUBJECT	AGE	FREQUENCY	ARM SWINING - NO FOV/SR		ARM SWINING - FOV		ARM SWINING - SR	
			DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT
1	24	x > 10	2	7	1	5	2	7
2	23	x > 10	1	6	3	5	1	7
3	24	1 > x > 10	5	3	4	5	5	4
4	25	1 > x > 10	6	4	6	6	6	6
5	23	x > 10	1	8	1	7	1	8
6	23	x > 10	4	10	6	9	3	10
7	23	0	1	9	1	9	1	9
8	27	1 > x > 10	5	6	4	6	5	6
9	24	1 > x > 10	8	9	9	7	8	8
10	23	0	7	6	7	6	7	6
11	22	0	1	9	2	9	1	9
12	29	1 > x > 10	1	9	1	7	1	9
13	24	0	5	7	4	6	4	7
14	22	0	5	7	4	7	4	7
15	27	x > 10	1	8	3	6	1	7

SUBJECT	AGE	FREQUENCY	FLYING - NO FOV/SR		FLYING - FOV		FLYING - SR	
			DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT	DIZZINESS	ENJOYMENT
1	24	x > 10	1	10	5	6	1	10
2	23	x > 10	2	9	10	6	2	10
3	24	1 > x > 10	4	9	4	9	4	9
4	25	1 > x > 10	10	1	10	1	10	1
5	23	x > 10	1	9	1	8	1	9
6	23	x > 10	7	10	6	10	6	10
7	23	0	1	10	1	10	1	10
8	27	1 > x > 10	2	10	2	10	2	10
9	24	1 > x > 10	3	10	5	8	2	10
10	23	0	1	10	1	10	1	10
11	22	0	2	10	1	10	2	10
12	29	1 > x > 10	3	1	2	10	2	10
13	24	0	5	3	5	8	4	9
14	22	0	3	10	3	9	3	10
15	27	x > 10	1	10	1	10	1	10

Annex III – Propuesta TFG:

1. TÍTULO DEL PROYECTO

Diseño e Implementación de un Sistema de Navegación en Entornos de Realidad Virtual para Videojuegos Comerciales.

2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TEMA A TRATAR

La cada vez mayor relevancia de los dispositivos de realidad virtual en entornos profesionales y domésticos evidencia la necesidad de mejorar los actuales sistemas de control y crear soluciones más eficaces y mejor adaptadas al ser humano. Este proyecto se centra en la creación de un sistema de navegación que, mediante el análisis de aproximaciones previas y de las necesidades de los usuarios en el entorno práctico de los salones recreativos, sirva para estudiar posibles vías de mejora en cuestiones como la precisión, la percepción espacial y el equilibrio.

3. OBJETIVOS DEL PROYECTO

Los objetivos de este proyecto son:

1. Analizar los métodos de control y desplazamiento más comunes en entornos de realidad virtual.
2. Estudiar sistemas y propuestas innovadoras existentes como posibles vías de estudio.
3. Diseñar e implementar un sistema de navegación para su uso en salones recreativos que permita desplazarse por terrenos y simular el vuelo.
4. Analizar con usuarios el funcionamiento de los sistemas tradicionales y el sistema ideado en el proyecto y comparar los resultados de dichas pruebas.

4. METODOLOGÍA

La metodología se establecerá en las primeras fases del proyecto.

5. PLANIFICACIÓN DE TAREAS

Las tareas quedan predefinidas de manera global en los objetivos. Serán fijadas de forma concreta durante el desarrollo del proyecto.

Annex IV - Tutorías

1. **Fecha:** Miércoles 25 de Septiembre de 2019. **Modo:** Presencial. **Lugar:** Universidad San Jorge, Zaragoza.

Contenido: En esta primera toma de contacto le propuse a Daniel Blasco que fuera mi tutor para el Trabajo de Fin de Grado de Videojuegos. Tras aceptar, nos reunimos para tener una pequeña charla en la que le expuse las bases de mi idea, es decir, un estudio sobre la Cinetosis en Realidad Virtual con sujetos de pruebas. El me dio consejos de cómo llevar a cabo un estudio así para que empezara a prepararlo.

Decimos volver a reunirnos en Octubre para darle el visto bueno a la propuesta del TFG y centrar los objetivos del estudio.

2. **Fecha:** Miércoles 27 de Octubre de 2019. **Modo:** Online. **Lugar:** Webmail.

Contenido: Le envió a Daniel mi propuesta inicial del Trabajo de Fin de Grado. Mediante el intercambio de unos pocos correos electrónicos terminé de redactar la propuesta gracias a sus consejos y la envió.

Terminamos el intercambio de correos electrónicos diciéndole yo que voy a presentarme directamente a la convocatoria de Septiembre para poder centrarme en el resto de asignaturas de la carrera. Acordamos continuar en Junio.

3. **Fecha:** Martes 7 de Julio de 2020. **Modo:** Online. **Lugar:** Discord.

Contenido: En esta reunión hablamos de los cambios que mi Trabajo ha sufrido por la crisis del Covid-19. Tras solucionarlo, le expongo a grandes rasgos la metodología y proceso de trabajo.

Nos intercambiamos disponibilidad para tutorías durante el verano, acordando por mi parte mandarle pequeños informes sobre el estado actual del desarrollo, para mantenerlo informado del avance y para avisarle sobre la necesidad de futuras tutorías.

4. **Fecha:** Lunes 24 de Agosto de 2020. **Modo:** Online. **Lugar:** Discord.

Contenido: Solicito tutoría para enviar a Daniel la primera versión completa de mi Trabajo de Fin de Grado. Tras leerlo, me da un par de consejos de cómo mejorar un par de secciones que podrían dar lugar a dudas para alguien que no entienda cierta terminología que empleo. A su vez, le informo del problema de presentar mi build al jurado siendo que es de Realidad Virtual y la presentación puede que no sea presencial, le pregunto si puedo tener a un ayudante que se ponga las gafas por mí para mostrar la demo al jurado sin tener que apartarme del ordenador, a lo cual no sabe decirme si sería posible o no dadas las circunstancias actuales.

Se acuerda seguir trabajando en el proyecto para darle los últimos retoques, así como ponerme en contacto con Jorge Echeverría para preguntarle lo del ayudante.

Bibliography

- [1] VIRTUIX. *Virtuix Omni Platform* [Online]. Virtuix Store. Available from: <https://www.virtuix.com/product/virtuix-omni/>
- [2] DOBIE, THOMAS G. *What Is Motion Sickness?* [Online]. Science Beta. March 2018. Available from: <https://sciencebeta.com/motion-sickness/>
- [3] RHINOX. *Motion sickness in VR and what we do to minimize the effects* [Online]. Rhinox. June 2020. Available from: <https://www.rhinox.be/blog/motion-sickness-in-vr>
- [4] HTC. *HTC Vive Trackers* [Online]. Vive. Available from: <https://www.vive.com/us/accessory/vive-tracker/>
- [5] INTELLECTSOFT. *Motion Sickness In Virtual Reality: How Big Is the Problem And How Good Are the Solutions* [Online]. IntellectSoft. July 2015. Available from: <https://www.intellectsoft.net/blog/motion-sickness-in-virtual-reality-how-big-is-the-problem-and-how-good-are-the-solutions/>
- [6] WIKIPEDIA. *Field of view in video games* [Online]. October 2018. Available from: https://en.wikipedia.org/wiki/Field_of_view_in_video_games
- [7] SKIPCLARKE. *Screen change* [Online]. WSFG.ORG. August 2008. Available from: <https://www.wsgf.org/article/screen-change>
- [8] JERALD, Jason. *The VR Book: Human-centered Design for Virtual Reality*. Association for Computing Machinery Eight Volume. New York (NY) US: Association for Computing Machinery, 2016. 599 p. ISBN: 1970001127, 9781970001129
- [9] WIKIPEDIA. *Perspective distortion (photography)* [Online]. June 2019. Available from: [https://en.wikipedia.org/wiki/Perspective_distortion_\(photography\)](https://en.wikipedia.org/wiki/Perspective_distortion_(photography))
- [10] BOND, David; NYBLOM, Madelein. *Evaluation of four different virtual locomotion techniques in an interactive environment*. Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden, 2019. Available from: <https://www.diva-portal.org/smash/get/diva2:1334294/FULLTEXT02>

-
- [11] CASTILLO, José C. *4FreaksFiction mostrará 'H.A.T.' en la Indie Zone de Fun & Serious Game Festival* [Online]. El Correo. October 2018. Available from: <https://www.elcorreo.com/tecnologia/fs-gamer/eventos/fs-game-festival/4freaks-fiction-mostrara-20181030113719-nt.html?ref=https:%2F%2Fwww.google.com%2F>
- [12] STATT, Nick. *How game designers find ways around VR motion sickness* [Online]. The Verge. October 2016. Available from: <https://www.theverge.com/2016/10/13/13261342/virtual-reality-oculus-rift-touch-lone-echo-robo-recall>
- [13] WHITTINGHILL, David Matthew , et al. *Nasum virtualis: A simple technique for reducing simulator sickness*. West Lafayette, Indiana, US : s.n., 2015.
- [14] ROETTIGERS, Janko. *The VR gaming centers of the future may not survive the crisis of today* [Online]. Protocol. June 2020. Available from: <https://www.protocol.com/location-based-vr-covid-19>
- [15] HTC. *Vive Cosmos Headset* [Online]. Vive. Available from: <https://www.vive.com/mx/product/vive-cosmos/features/>
- [16] UNITY TECHNOLOGIES. *Unity* [Online]. Unity. Available from: <https://unity.com/>
- [17] ALLEN, David. *Getting Things Done: The Art of Stress-Free Productivity*. City of Westminster, London, England. Penguin Books. 2001. 267 p. ISBN: 978-0-14-312656-0V.
- [18] VALVE CORPORATION. *SteamVR Plugin* [online]. V2.6.1. Unity Asset Store. August 2020. Available from: <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
- [19] UNITY TECHNOLOGIES. *Unity Asset Store*. [Online]. Available from: <https://assetstore.unity.com/>
- [20] ADOBE. *Mixamo* [Online]. San Francisco 2018. Available from: <https://www.mixamo.com/>
- [21] CAFOFO. *Sci-Fi Sound Pack* [Online]. CafofoMusic Available from: <https://cafofomusic.com/scifi-sound-pack>

- [22] SYNTY STUDIOS. *POLYGON - Sci-Fi City Pack* [Online]. Synty Store. Available from: <https://syntystore.com/products/polygon-sci-fi-city>
- [23] SYNTY STUDIOS. *POLYGON - City Pack* [Online]. Synty Store. Available from: <https://syntystore.com/collections/frontpage/products/polygon-city-pack>
- [24] SYNTY STUDIOS. *POLYGON - Heist Pack* [Online]. Synty Store. Available from: <https://syntystore.com/products/polygon-heist-pack>
- [25] HP. *HP Z1 Entry Tower G5* [Online]. HP Store. Available from: <https://store.hp.com/us/en/pdp/hp-z1-entry-tower-g5-p-8ag71ut-aba-1>
- [26] ACER. *Acer Full HD (1920 x 1080) IPS Ultra-Thin Zero monitor* [Online]. Amazon. Available from: https://www.amazon.com/-/es/Acer-SB220Q-Ultra-Thin-Frame-Monitor/dp/B07CVL2D2S/ref=sr_1_2?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=monitor&qid=1597936970&sr=8-2
- [27] VICTSING. *VicTsing Wireless Mouse and Keyboard* [Online]. Amazon. Available from: https://www.amazon.com/-/es/VicTsing-KeyBoard-Adjustable-Independent-Indicator/dp/B07TT3VN4X/ref=sr_1_5?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=DIYG9P3ZQ34D&dchild=1&keywords=mouse+and+keyboard&qid=1597937010&srefix=mouse+and+key%2Caps%2C239&sr=8-5
- [28] HTC. *VIVE Cosmos* [Online]. PCComponentes. Available from: <https://www.pccomponentes.com/htc-vive-cosmos-gafas-de-realidad-virtual>
- [29] INDEED. *Salarios para empleos de Desarrollador/a de software en España* [Online]. Indeed. August 2020. Available from: <https://es.indeed.com/salaries/desarrollador-de-software-Salaries?period=hourly>
- [30] FEINER, Steven K.; FERNANDES, Ajoy S. *Combating VR sickness through subtle dynamic field-of-view modification*. Greenville, SC, USA. IEEE Symposium on 3D User Interfaces (3DUI) 2016. ISBN: 978-1-5090-0842-1.

-
- [31] BUCKLE, Chase. AR vs VR: *The Challenges and Opportunities in 2019* [Online]. Global Index. November 2019. Available from: <https://blog.globalwebindex.com/chart-of-the-week/augmented-virtual-reality/>
- [32] ARVILAB. *Combating VR sickness: debunking myths and learning what really works* [Online]. ArviVR. March 2018. Available from: <https://vr.arvilab.com/blog/combating-vr-sickness-debunking-myths-and-learning-what-really-works>
- [33] UNITY TECHNOLOGIES. *Navigation and Pathfinding* [Unity]. Unity User Manual (2020.1). August 2020. Available from: <https://docs.unity3d.com/2020.1/Documentation/Manual/Navigation.html>
- [34] UNITY TECHNOLOGIES. *Cinemachine* [Unity]. Art and Design. 2020. Available from: <https://unity.com/es/unity/features/editor/art-and-design/cinemachine>
- [35] UNITY TECHNOLOGIES. *Universal Render Pipeline* [Unity]. Unity Packages (2020.1). 2020. Available from: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@8.2/manual/index.html>
- [36] UNITY TECHNOLOGIES. *Rigidbody* [Unity]. Unity User Manual (2020.1). August 2020. Available from: <https://docs.unity3d.com/es/2020.1/Manual/class-Rigidbody.html>
- [37] UNITY TECHNOLOGIES. *Capsule Collider* [Unity]. Unity User Manual (2019.4). August 2020. Available from: <https://docs.unity3d.com/es/2020.1/Manual/class-CapsuleCollider.html>
- [38] UNITY TECHNOLOGIES. *ParticleSystem* [Unity]. Unity User Manual (2020.1). August 2020. Available from: <https://docs.unity3d.com/ScriptReference/ParticleSystem.html>
- [39] WIKIPEDIA. *Shader* [Online]. June 2019. Available from: <https://en.wikipedia.org/wiki/Shader>
- [40] MANCHAISER. *Highlight Shader for SteamVR using Universal Render Pipeline* [GitHub]. ValveVR GitHub Forum. June 2019. https://github.com/ValveSoftware/steamvr_unity_plugin/issues/422#issuecomment-502504360

- [41] UNITY TECHNOLOGIES. *Transform.Translate* [Unity]. Unity User Manual (2019.4). August 2020. Available from: <https://docs.unity3d.com/ScriptReference/Transform.Translate.html>
- [42] UNITY TECHNOLOGIES. *Time.deltaTime* [Unity]. Unity User Manual (2019.4). August 2020. Available from: <https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>
- [43] UNITY TECHNOLOGIES. *Rigidbody.AddForce* [Unity]. Unity User Manual (2019.4). August 2020. Available from: <https://docs.unity3d.com/ScriptReference/Rigidbody.AddForce.html>
- [44] CORBERTO. *How to climb stairs as a Rigidbody* (in Unity3D) [Online]. Corbertos Blog. September 2015. <https://cobertos.com/blog/post/how-to-climb-stairs-unity3d/>
- [45] STILL NOT SURE. *Airborne* [Online]. Steam. July 2020 Available from <https://store.steampowered.com/app/1010100/Airborn/>
- [46] KITAOKA, Akiyoshi. *Differences in perceived speed 2* [Online]. Twitter. February 2020 Available from <https://twitter.com/AkiyoshiKitaoka/status/1224692520283598854>