FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Physics-based platform for Collaborative Interaction in Virtual Reality

Carlos Miguel Ferreira Lucas



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Jacob

Co-Supervisor: Rui Nóbrega

July 25, 2020

Physics-based platform for Collaborative Interaction in Virtual Reality

Carlos Miguel Ferreira Lucas

Mestrado Integrado em Engenharia Informática e Computação

July 25, 2020

Abstract

One of the areas of application of Virtual Reality (VR) is the possibility of representing and interacting with a reality in which a set of situations can be represented but that otherwise could not be experienced. In this way, VR interaction techniques provide a set of tools that can be applied to facilitate and simplify the representation of a certain context that is intended to be understood.

Likewise, considering the complexity that involves physical laws and their actions on objects, the application of interaction and representation techniques in VR may provide a means of study for simulating the interaction with a given object based on the physics involved.

Thus, the objective of this work is to develop a platform that allows a VR remote interaction with an object and visualize its behavior in the represented scenario. This interaction will allow the user to predict and simulate the effects on the object, by varying some of the parameters associated with the physical laws that allow to represent this behavior.

A set of techniques were analyzed and implemented in order to obtain parameters that allow to perceive which is the one that best suits a given action, taking into account the representation of the action itself, the ease of use of that interaction and its degree of immersiveness.

After this analysis, it was decided to develop a platform that allows remote interaction with a basketball through its representation in a hologram and including some specific details such as the representation of the ball's own trajectory towards the basketball hoop and the possible influence of velocity and gravity in this trajectory.

In this platform, details regarding the user's own immersion during the experience were taken into account, as well as an asymmetric collaboration component that allows the use of the platform by two users simultaneously, each having a different role in the scenario in question, but with both having to reach a common objective.

Finally, studies were carried out with some users in order to assess the quality of the developed platform. It was also considered the possibility of allowing the created scenario to be adapted to represent other contexts.

Keywords: Virtual Reality, Human-Computer Interaction, Remote Interaction, Asymmetric Collaboration

Resumo

Um dos âmbitos de aplicação da Realidade Virtual (RV) é a possibilidade de representar e interagir com uma realidade na qual podem estar representadas um conjunto de situações reais mas que de outra forma não poderiam ser experienciadas. Desta forma as técnicas de interação de RV disponibilizam um conjunto de ferramentas que podem ser aplicadas para facilitar e simplificar a representação de um determinado contexto que se pretende dar a entender.

De igual modo, considerando a complexidade que envolve as leis físicas e a sua ação sobre os objetos, a aplicação de técnicas de interação e representação em RV poderão proporcionar um meio de estudo para simulação a interação com um determinado objecto com base na física envolvida.

Assim, o objetivo deste trabalho é desenvolver uma plataforma que permita em RV a interação remota com um objeto e a visualização do seu comportamento no espaço representado. Esta interação irá permitir ao utilizador prever e simular os efeitos sobre o objeto, ao fazer variar alguns dos parâmetros associados às leis físicas que permitem representar este comportamento.

Neste sentido, foi analisado e implementado um conjunto de técnicas de forma a obter indicadores que permitam perceber aquela que melhor se adequa a uma determinada ação, tendo em conta: a representação da ação em si, a facilidade de utilização dessa interação e o seu grau de imersividade.

Após esta análise, optou-se por desenvolver uma plataforma que permite a interação remota com uma bola de basquetebol através da sua representação num holograma e incluindo alguns detalhes específicos como a representação da própria trajetória da bola em direção ao cesto e da possível influência da velocidade e da gravidade nesta mesma trajetória.

Nesta plataforma foram tidos em conta os detalhes relativos à própria imersão do utilizador durante a experiência, bem como foi adicionada uma componente de colaboração assimétrica que permite a utilização da plataforma por dois utilizadores em simultâneo, tendo cada um deles uma função diferente no cenário em causa, mas em que ambos têm um objetivo comum.

Por fim, foram realizados estudos com alguns utilizadores de modo avaliar a qualidade da plataforma desenvolvida. Foi também tida em conta a possibilidade de permitir que o cenário criado possa ser adaptado para a representação de outros contextos.

Acknowledgements

To my supervisors, Rui Nóbrega and João Jacob for always providing relevant feedback and guiding me throughout this process, while also motivating me to do it.

To my parents, for being supportive and for cheering for me at all time.

To my sister, for always taking the role of "big sister", and lecturing me whenever I stepped the wrong way.

To Hugo, for all that we have accomplished, and all that we still will.

To you Rita, for always being there with me every step of the way, and not letting me lose my focus, while constantly reminding me of how strong you are. Thank you for your patience and understanding.

Finally, to you Conceição. Thank you for your devotion, for your firm belief and unconditional support. Don't feed Leonor too much.

Miguel Lucas

"Do not go gentle into that good night.
Rage, rage against the dying of the light."

Dylan Thomas

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Problem . . . 2
- 1.3 Research Questions . . . 3
- 1.4 Objectives . . . 3
- 1.5 Document Structure . . . 3

2 State of the Art Review **5**
- 2.1 Virtual Reality Environments . . . 5
  - 2.1.1 VR Devices . . . 6
  - 2.1.2 Virtual World Interaction . . . 8
  - 2.1.3 Movement/Locomotion Techniques . . . 13
- 2.2 Computer-based Physics Simulation Visualizers . . . 17
  - 2.2.1 Dedicated software . . . 18
  - 2.2.2 Related software . . . 19
  - 2.2.3 Game Engines . . . 20
- 2.3 Science Simulators and VR applications . . . 20
  - 2.3.1 Medicine . . . 21
  - 2.3.2 Education and Training . . . 22
  - 2.3.3 Entertainment and Culture . . . 22
  - 2.3.4 Engineering . . . 23
- 2.4 Summary . . . 25

3 Requirements and Architecture **27**
- 3.1 Problem Analysis . . . 27
- 3.2 Requirements . . . 28
- 3.3 System Architecture . . . 29
- 3.4 Usability . . . 30
- 3.5 Summary . . . 32

4 Implementation **33**
- 4.1 Case Study . . . 33
- 4.2 Tools and Technologies . . . 36
- 4.3 Project interface . . . 37
- 4.4 VR Interaction/Locomotion Techniques . . . 44
- 4.5 Physics simulation . . . 47
  - 4.5.1 Hologram Manipulation . . . 48
  - 4.5.2 Ball Trajectory . . . 49

4.5.3 Time Control . . . 50
4.5.4 Wind . . . 50
4.5.5 Shot detection . . . 51
4.6 Collaboration and Networking . . . 51
4.6.1 Asymmetric Collaboration . . . 51
4.6.2 Sending and Receiving Signals . . . 53
4.7 External Settings . . . 54
4.7.1 Gravity . . . 55
4.7.2 Wind . . . 56
4.8 Retrieving analytics data . . . 56
4.9 Summary . . . 58

5 Evaluation **59**
5.1 Evaluation Protocol . . . 59
5.2 Test Session Structure . . . 61
5.3 Available Data . . . 62
5.3.1 User questionnaires . . . 63
5.3.2 Experiment Usage Data . . . 63
5.4 User Studies . . . 64
5.4.1 User Population overview . . . 64
5.4.2 Single-player setting . . . 64
5.4.3 Collaborative setting . . . 73
5.5 Results Analysis and Discussion . . . 80
5.6 Summary . . . 83

6 Conclusion and future work **85**
6.1 Future Work . . . 86

References **87**

A Experiment Data Example **93**

B Consent Form **97**

List of Figures

2.1 Pointer to rotate the cylinder, in the VRTK Framework[1] . . . 10
2.2 Occlusion Selection technique, without (left) and with (right) vertical offset [AA13] 10
2.3 Selecting an object with gaze[2] . . . 11
2.4 Variants of VR menu display, based of Kim et al. [KKP+00] . . . 13
2.5 Menu interaction in the Oculus Quest . . . 13
2.6 A user walking virtually straight while being redirected [LS18] . . . 15
2.7 Point and teleport technique, implemented using the VRTK framework[3] . . . 17
2.8 The world-in-miniature technique, showcase a miniature of the room the user is currently in [SCP13] . . . 17
2.9 The Projectile Motion with Angry Birds project, using OSP[7] . . . 19
2.10 A neurosurgery resident training in the NeuroTouch prototype [DLDM12] . . . 21
2.11 Steam total users playing with a VR headset. (the seven months gap is due to erroneous data from a Valve collection issue)[4] . . . 23
2.12 Volvo XC90 Virtual Test Drive[5] . . . 24

3.1 General System Architecture . . . 30

4.1 Physics problem, requesting to calculate the friction force on the sliding block. [Vil19] . . . 34
4.2 Various anchor points defined in the video game Poly Bridge.[6] . . . 35
4.3 Basketball court, board and ball. . . . 38
4.4 A hologram to represent an elephant in Circus Roncalli.[7] . . . 38
4.5 Hologram of the original ball. . . . 39
4.6 Representation of the trajectory curve, including bounces. . . . 40
4.7 Detail of the axes and the velocity text cues. . . . 40
4.8 Current velocity vector (at yellow) and textual cue. . . . 41
4.9 Red and blue guidelines around the edge of the basketball court. . . . 41
4.10 Simple effect for when a user scores a shot. . . . 42
4.11 Different panels of the scoreboard. . . . 43
4.12 Leaves blown to indicate the wind direction. . . . 43
4.13 The direction of the leaves will be randomly selected, but limited to a direction that is inside the cone. . . . 44
4.14 Using the time control will spawn the blue trajectory, representing the course the ball will have if the motion is continued. . . . 44
4.15 Steps of grabbing the hologram. . . . 45
4.16 Forward joystick movement. The user will move in the direction of the green arrow.[8] 46
4.17 Using the teleportation technique. . . . 47
4.18 Different perspectives of the same hologram vector, giving the same result. . . . 48

4.19 Trajectory makes it look like it will be a successful shot, but the ball will actually hit the side of the rim and bounce out. 50
4.20 Shot classification areas. 52
4.21 Perspectives of the same shot for the manipulator and the observer. 53
4.22 Three coloured buttons, for the observer to send signals to the manipulator. . . . 54
4.23 Three snapshots, sent from the observer. Two red and one green. 54

5.1 The distance parameter is the difference between the point at the center of the basketball and the center of the basketball hoop.[9] 64
5.2 Single-player Questions SP.1 and SP.2 66
5.3 Single-player Questions SP.3 and SP.4 66
5.4 Single-player Question SP.5 67
5.5 SP-DBS number of grabs and time spent per shot 69
5.6 SP-DBS hit percentage 69
5.7 SP-VBS number of grabs and time spent per shot 70
5.8 SP-VBS hit percentage 71
5.9 SP-TCS number of grabs, time spent per shot and distance to target 72
5.10 SP-TCS hit percentage 72
5.11 Multiplayer Questions MP.1 and MP.2 75
5.12 Multiplayer Questions MP.3 and MP.4 75
5.13 Multiplayer Questions MP.5 and MP.6 75
5.14 MP-DBS number of grabs and time spent per shot 77
5.15 MP-DBS hit percentage 78
5.16 MP-DBS number and type of signals received per shot position 78
5.17 MP-TCS number of grabs, time spent per shot and distance to target 81
5.18 MP-TCS hit percentage 81
5.19 MP-TCS number and type of signals received per shot position 82

List of Tables

2.1 Most popular VR HMDs 7
2.2 VR Interaction Techniques Overview 25

4.1 Scoring all scenarios on the defined parameters 36

5.1 Tasks overview and number of shots taken per task 61
5.2 Structure of each test session 62
5.3 Participants' Gender 65
5.4 Participants' Age 65
5.5 Participants' experience with VR 65
5.6 Summary of the results of SP.1 to SP.5 66
5.7 Summary of the results of MP.1 to MP.6 76

Abbreviations

HCI	Human-computer Interaction
VR	Virtual Reality
HMD	Head Mounted Display
WIMP	Windows, Icons, Menus, Pointers
DoF	Degrees of Freedom

Chapter 1

Introduction

This chapter will present the context and motivation behind the dissertation, the problem and research questions addressed and the objectives for this research. Finally, the document structure is explained to serve as a guide for the related topics.

Interaction between humans and computers has always been a key factor when developing applications. This human-computer interaction (HCI) is present in our everyday life; for example, when we check our email, the ones we haven't read yet are displayed in bold. This isn't a decision made by choice, but by years of researching that came to a conclusion that it was the best way to catch human attention to something that might be important. How we then interact with it, whether in our mobile phones or computers is constantly evolving, and needs a constant research effort to keep track of new and better ways for humans to interact with a computer-based system. In sum, HCI research is about solving problems related to the human use of computing [OH16].

In particular, Virtual Reality (VR) has shown its potential since it introduced new ways to represent a known reality and interact with that reality in this different context. Although scenes such as flying a jet or tank, are already tangible, another scenes, such as being able to feel the dry air of the Sahara in geography classes or feel the hard, cold scales of a dragon in a computer game seem to be in a long way from now [Boa12]. However, the applications for VR scenarios are uncountable and may be focused on fictional concepts or more real ones, although all of these representations are based in multiple human-interactions which by itself are limited or at least influenced by some human conditions. These conditions goes from physical sickness during the application usage to concepts misrepresented which may decreases the immersion factor or the human processing of these concepts [RTBG17].

These factors are particularly relevant while applying VR to the representation of physical laws and its assimilation/representation for learning purposes.

1.1 Motivation

The representation of some physics subjects and its complex details are one the multiple areas where some VR interaction techniques may help to create a positive improvement. The interaction and the visualization of its outcomes may also allow a better understanding of the different constraints and variables involved. The aim of this work is to create a platform that allows the representation of an object and the physics constraints related to this object, by creating a scenario that allows a remote interaction with this object while experiencing some of the circumstances that affect its behaviour in the environment, such as the example of the gravity and wind force.

This platform will be used not only for this thesis purpose but also to support the customization of new scenarios for testing and draw out new conclusions in such a way it could be done by someone with a less technical profile, for example, physics teachers.

1.2 Problem

Even though many VR interaction techniques have been categorized and implemented, there still is a need to research which actions best apply to these techniques. Some applications, like the video game Fallout 4 VR[1] offer different ways to handle movement or interaction, asking the users to select which technique they prefer. While this gives the users freedom of choice to align the game to their personal preferences, it increases the work load of the development and quality assurance teams, since they have to implement many different techniques. Even though big projects like Fallout 4 VR have the assets to do so, smaller projects may not have them, and may not have the time to test every technique to see which one fits their project the most.

Also, the fact that there is no universally accepted technique for a specific action, like movement, makes it that this becomes a research topic for every project in VR with user interaction. Providing these projects with a sandbox tool in order to conduct their own tests becomes a must-have, in order to facilitate the development and implementation.

However, there are also situations where remote interaction is preferred over a direct manipulation of objects, and providing a way to visualize and anticipate the interaction can prove to be a great asset in the overall experience. Building a car is one of these examples, as there are many pieces that compose it, and changing one of them remotely can help to perceive what happens to other components of the car from different perspectives other than the direct view of the changed piece itself.

A collaborative component allows the users to share their perceptions about the object's behaviour and complement the details that each one of them is experiencing and observing, even while assuming different roles in the same scenario.

[1] Fallout 4 VR video game, 2017: https://fallout.bethesda.net/en/games/fallout-vr

1.3 Research Questions

Following what has been stated as the problem, the following questions emerge to guide the research work to be approached by this document:

- Which VR **interaction** techniques have been researched and implemented?
- How to **remote control** an object without direct manipulation?
- How to combine these techniques to create a **collaborative** physics laboratory?

The first question aims to research, identify and categorize which VR interaction techniques have been developed. The second question aims to establish a way to control an object at a distance, without compromising the sense of control or immersiveness, and its advantages in certain types of interaction. The last question focuses on finding a way to combine the identified techniques to create a physics lab, in which these techniques can be explored in a collaborative environment.

1.4 Objectives

One of the main objectives of this work is to identify and implement different VR interaction techniques, focused on remotely controlling and manipulating an object. Later, using these techniques, studies will be done to show if these techniques attain a degree of effectiveness in such a way that it improves the interaction paradigm.

Another objective is to provide a collaborative case study example, where users can cooperate to reach a common objective. It is intended to assess if this cooperation is indeed beneficial to reach an objective, or if the users do feel more motivated in doing so, rather that reaching it alone.

1.5 Document Structure

After this introductory chapter, this document contains three additional chapters:

- **State of the Art Review** (chapter 2) - In this chapter, a state of the art review is presented, stating diferent ways of interacting in a virtual environment and how to interact in a hardware point of view;how physics has been applied to computer-based simulators;and which relevant areas has VR been used for.
- **Requirements and Architecture** (chapter 3) - In this chapter, an overview of the problem at hand is given, along with the necessary requirements and the proposed system architecture. Finally a discussion about the usability principles to be considered for the development of the prototype is presented.

- **Implementation** (chapter 4) - This chapter aims to explain the implementation of this work, starting with the case study implemented, and the reasons why it was chosen. It also provides insights on various interface components, implemented interaction techniques and collaboration aspects involved.
- **Evaluation** (chapter 5) - In this chapter, some methods to evaluate the effectiveness of the proposed solution are discussed, along with its results.
- **Conclusion and future work** (chapter 6) - A conclusion of what this document has proposed is presented, along with some suggestions to where this work could go for in the future.

Chapter 2

State of the Art Review

This chapter will provide an overview of the areas related with the topics of this dissertation: virtual reality environments and how to interact with them, in a hardware level, and also on a more technical point-of-view; physics-based applications; and uses of VR for the sake of science, while also analyzing relevant projects that can contribute to this work. Firstly, an introduction to which VR devices are currently available will be presented, with their advantages and disadvantages. The next sections will focus on VR interaction and locomotion techniques, categorizing them according to their usage. Then, a discussion on computer-based physics simulations and how to do them will take part. Finally, an overview of projects that are using VR for their advantage will be conducted.

2.1 Virtual Reality Environments

Following the same logic as video games, virtual reality environments as a concept is a representation of another reality but where the user can interact with the environment, reason why it differs from other ways of representing a fictional reality, like movies. Virtual reality environments also differ from a common concept of a video game since a virtual reality environment creates a user experience that groups different kinds of sensations and interactions. This fundamental aspect of VR delivers an experience that gives rise to an illusory sense of place and an illusory sense of reality that distinguishes it fundamentally from all other types of media [SSV16]. VR environments may create in the user the complex sensation of vertigo, far more closer to the real sensation, taking advantage of the conjugation of sound effects with visual effects like sounds in the user's back and the possibility to rotate the head by some degrees and notice the change of the sound effect.

VR environments are being developed in order to include such more complex and different interactions and details. In the '60s, in what is depicted as one of the first examples of virtual reality, the Sensorama displayed a movie of a motorcycle running through Brooklyn, giving a high

amount of feedback to make the user feel like he is actually there. There was, however, no interaction with the movies [Boa12]. Almost 60 years later, and now it's possible to have an experience much closer to reality, like being underwater or flying which is based in different psychological and physical details to create this sensation in the user's mind. This individual's involvement with the virtual environment due to objective, stimulating conditions is referred to as **immersion**. The virtual environment's visual, auditory, and tactile designs create the three-dimensional perception that the VR model is the real world. The subjective experience of physically being in a virtual environment and that this environment is real is referred to as **presence**. Characteristics of presence are the perception of the environment being real, blanking out real-world stimuli as well as involuntary and objectively meaningless body movements. For example, a person might crouch to feel his feet firmly planted on the real floor while crossing a virtual bridge over a virtual abyss. [Eic10]

Virtual reality profits from the exploitation of the brain to produce illusions of perception and action. This is like finding loopholes in the brain's representations and then making use of these to produce an illusory reality [Sla14]. The diversity of the immersing factors, as well as the combinations of sensations that could be planned for the user to feel during the VR experience can vary according to the device being used and its characteristics, which will be explained in the following subsection.

2.1.1 VR Devices

As it happens with some other technologies, the equipment available for VR is being developed in different directions. There are some devices that could be attached to a smartphone, other to a PC, and there are also standalone structures. Also, the rise of commercial HMDs helped to stimulate continuous improvement of the technology, with new equipment and techniques surging every year.

The different equipment and accessories that influence the interaction with the virtual environment will be discussed in this subsection.

2.1.1.1 Keyboard and mouse

The use of keyboard and mouse is currently the most commonly used I/O devices for interacting with desktop applications, but the same is not applied to VR applications. With the appearance of commercial HMDs like the Oculus Rift and the HTC Vive, new and VR-specific designed interaction devices appeared that surpasses the use of a keyboard and mouse in a VR application. Nevertheless, it's still possible to use these to interact in a virtual environment as Robertson et al. [RCM93] suggests that it is an inexpensive way to interact in non-immersive VR environments.

2.1.1.2 Head-mounted displays

The release of the Oculus Rift DK1 in 2013 is considered a landmark in modern VR, as it opened the consumer market to VR [BCK17]. Since then, Head-mounted displays (HMDs) like the Oculus Rift have appeared and are the preferred way of interacting with a virtual world. Most HMDs have two LCD displays, one for each eye, creating a stereoscopic image that is slightly different for each eye, but create the effect of depth for a 3D environment. In order to provide a fully immersive experience, the field of view is also a major concern, with most HMDs having between 100º to 110º field of view to match the human eyes field of view. These are combined with tracking sensors, that track the movement of the HMD (and therefore, the user's head) and replicates this movement onto the virtual world. A list of currently available HMDs is presented in Table 2.1.

Table 2.1: Most popular VR HMDs

HMD	Resolution	FoV	Framerate	Year	Controls
Oculus Rift	2160*1200	110º	90 FPS	2013	Oculus Touch controllers
HTC Vive	2160*1200	110º	90 FPS	2016	HTC Vive controllers
HTC Vive PRO	2880*1200	110º	90 FPS	2019	HTC Vive controllers
Valve Index	2880*1600	130º	120 FPS	2019	Valve Knuckles
Playstation VR	1920*1080	100º	120 FPS	2016	Playstation controllers
Asus Windows Mixed Reality (WMR)	1440*1440	95º	90 FPS	2017	WMR Motion Controllers
Google Cardboard	Same as smartphone	90º	60 FPS	2014	Side button for screen press
Samsung Gear VR	Same as smartphone	101º	60 FPS	2017	Bluetooth controller

2.1.1.3 Hand-tracking and Controllers

In order to allow a greater sense of immersion and interaction with a VR system, a way to manipulate the virtual world is needed. This can be either through gaze controls, where the action takes place inside the user's field of view, which is more common in mobile VR applications, or through controllers/gamepads. Specific controllers, like the HTC Vive controllers or the Oculus Touch are designed to operate specifically for a VR experience. These controllers are tracked through motion sensors and work as if they were an extension of the hand, allowing the user to see them in the virtual world (they can also be substituted by a different 3D model, like a hand). Nevertheless, more traditional controllers can be used like the Playstation or XBox controllers, but will provide a lesser sense of immersion and presence.

Another possibility is through the use of hand tracking devices. These devices track the user's real hand movements through computer vision techniques, and replicate the movements of the

hand into the virtual world. Inside the virtual world, the hand's avatar act as if they were a controller, and allows the user to interact with virtual objects. Devices such as the Leap Motion[1] allows for such capabilities and can generally be connected to any type of HMD. Even though the prospect of using bare hands to interact with a virtual world may sound exciting, in the comparison of the HTC Vive controllers versus Leap Motion, Gusai et al. [GBSC17] found that using controller-based techniques wielded significant better results than hand-tracking for object interaction. Users reported feeling much more comfortable using the controller, and also that it allowed a greater simplicity of interaction.

Moreover, some controllers, like the Oculus Touch the Valve Index Knuckles have combined the two techniques, providing their own controllers with finger-tracking techniques that provide the user with more subtle ways of interacting in a virtual world.

2.1.2 Virtual World Interaction

Taking into account the human-computer interaction paradigm, VR has given new ways to interact with a computer program. Since the first VR devices appeared in the 1960's, numerous VR interaction techniques have been developed. In Sutherland's "Sword of Damocles" HMD, cited as the first HMD created [ADI17, SSV16, Mcl01], two images were computer-generated, one for each eye. The 2D images were computed and rendered with an appropriate perspective with respect to the position of each eye in the three-dimensionally described virtual scene [SSV16]. The displays were mounted in a frame, which additionally had a mechanism to continually capture the position and orientation of the user's head, and therefore gaze direction. Hence, as the head of the user moves, turns, or looks up and down, this information is transmitted to the computer that recomputes the images and sends the resulting signals to the displays. It also contained sounds and it was updated according to the position and navigation of the user [ADI17, SSV16]. This can be considered as one of the first attempts of VR interaction.

Mark Mine [Min95] has defined five fundamental categories of VR techniques, which will be addressed in this document:

- Movement
- Selection
- Manipulation and Scaling
- Menu Interaction

Movement and locomotion techniques will be addressed later in this document, in chapter 2.1.3.

[1] Leap Motion, https://www.leapmotion.com, last access July 2019

2.1.2.1 Selection

Selection techniques can be defined as those which define the target of the desired interaction [Min95]. Inside the Selection techniques, we can subdivide them into local, at-a-distance, gaze directed, and voice input techniques.

Local interaction techniques can be described as those used to select an object by moving a cursor (typically attached to the user's hand) until it is within the object's selection region (like a minimal bounding box). Once chosen, the object can be selected using some pre-defined signal such as a gesture, button press, or voice command [Min95].

- The **touch** technique is defined by Forsberg [FHZ96] as one that allows the participant to place a 3D cursor (representing the tracked hand) on or inside a target object in the virtual environment.
- The **grabbing** technique is a gesture (e.g., with a glove input device) or button click signals to the application to select the target object [FHZ96].
- The **go-go** technique was proposed by Poupyrev et al. [PBW96] in which it allows the user to stretch its virtual arm to select distant objects. While this allows the user to reach for objects out of their reach, the precision decreases as users move their hand further because their movements are magnified.

At-a-distance techniques are those in which the user can select an object that falls outside of the immediate reach of the user [Min95]. Also known as ray-casting techniques, they make use of a virtual light ray (laser) to select an object, with the ray's direction specified by the user's hand. The use of the light ray makes the selection and grabbing task easy, because the user is only required to point to the desired object [BH97].

- The **laser pointer** technique utilizes ray intersection to determine which object(s) to select [FHZ96]. In this technique, a user points his hand in the direction of the object pretended and presses a switch or a button to activate the laser. When the laser collides with an object, that object is selected. Some variants of this technique exist, such as instead of pressing a button, the user holds that button pressed, and the object colliding with the laser is only selected when the user lets go of the button. An example of this technique can be found in Figure 2.1 where a laser pointer is used to point at a cube, which triggers the cylinder above to rotate.
- The **spotlight** or **flashlight** technique is a variation of the laser pointer technique. It uses a conic shape instead of a ray to select the pretended object. [LG94] A cone sticks out of the user's hand or controller, increasing its width with the distance. Because all the objects within the conic selection volume may be selected, however, a disambiguation metric for choosing a single object from the set of candidates may be required. [FHZ96]

[2] VRTK Pointer Documentation, https://vrtoolkit.readme.io/docs/vrtk_pointer/, last access June 2019

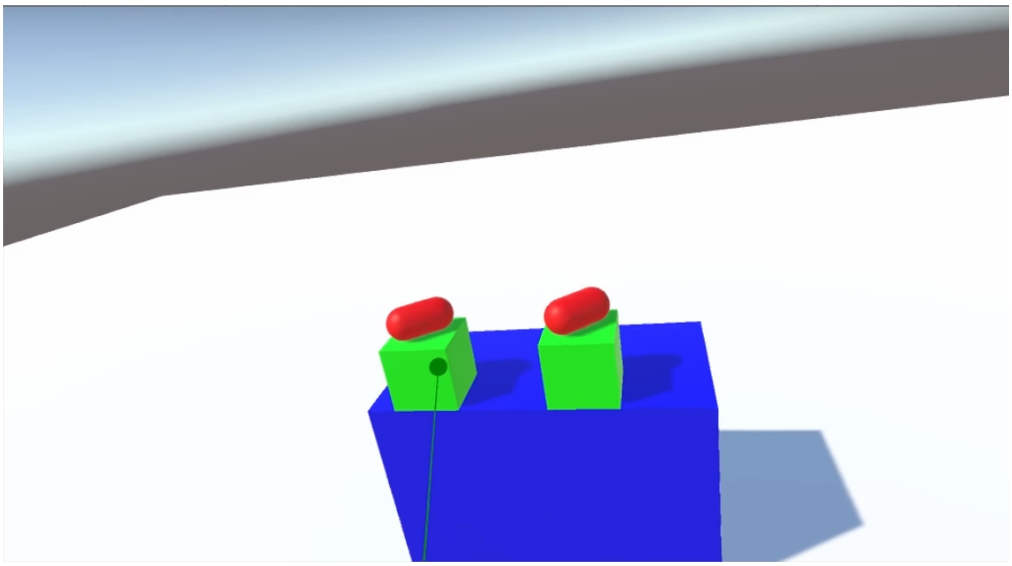


Figure 2.1: Pointer to rotate the cylinder, in the VRTK Framework[2]

- In **occlusion selection**, the pointing direction is defined by roughly aligning the hand with the eye position, thus requiring the user to keep its arm extended. However, this can be overridden by introducing a vertical offset that allows the user to keep his hand in a lower position, reducing fatigue levels [AA13], as can be seen in Figure 2.2.

Gaze-directed techniques are those which selection can be based upon the user's current gaze direction [Min95]; the user merely looks at an object to be selected and then indicates his selection via a button press. In other variations, the button press is substituted with a timer, and the user simply has to stare at the object long enough for the timer to reach 0. This type of techniques, seen in figure 2.3, has become widely popular for VR applications for mobile devices [KLJK17]. Atienza et al. [ABS+16] observed that the head gaze was effective as a means of control in the environment. Without much assistance, the players were even able to use this technique not only for object selection but also to navigate the virtual world, even though several participants reported nausea due to the sliding of the movement mechanism.

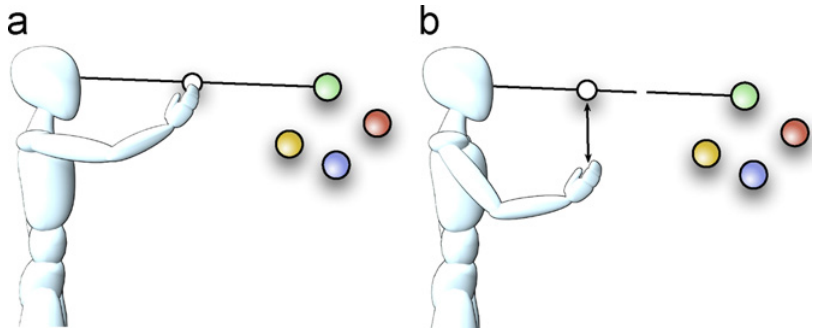


Figure 2.2: Occlusion Selection technique, without (left) and with (right) vertical offset [AA13]

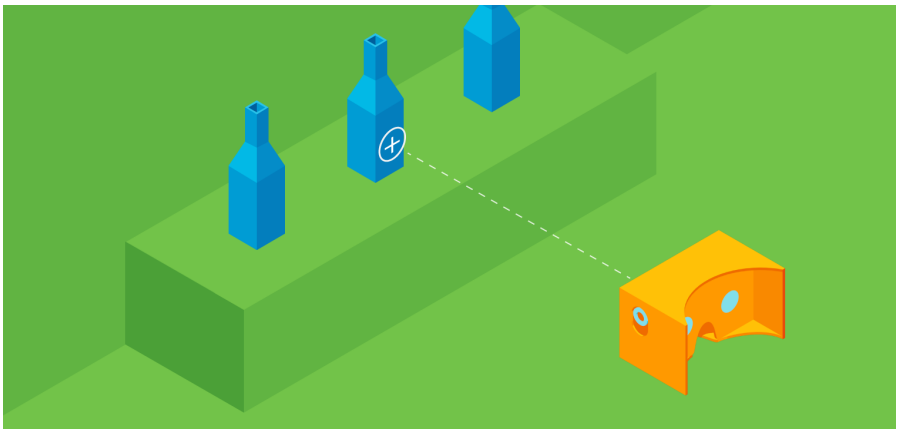


Figure 2.3: Selecting an object with gaze[3]

Voice input techniques can be used to identify objects to be selected and to signal actual selection. To select an object the user would issue a command such as "Select the Red Box" [Min95]. This command would then be interpreted internally, and the action would unfold. The main issues to be dealt with in such a system include the mental overhead of having to remember the names of all objects, specially in a virtual environment with a multitude of objects and increased clutter. A way to distinguish all objects would also be needed, as identical objects can cause conflicts in the process of selecting which one the user is referring to. The reliability of current voice recognition systems is also another obstacle.

Moreira et al. [MNR18] found that users with 3D content creation experience prefer this method of interaction instead of touch commands to manipulate a VR environment and create content on-the-fly.

2.1.2.2 Manipulation and Scaling

Object selection and positioning are among the most fundamental interactions between humans and environments, whether it is a “desktop” of 2D direct manipulation interface, 3D virtual environment, or the physical world [PBW96]. The interaction can either be realistic, where the user grabs the object and moves it as he would grab and move an object in the real world, or the user can manipulate objects in ways that would be impossible to represent in the real world [Min95].

Bowman [BH97] suggests that these techniques fall into 2 categories: arm-extension techniques and ray-casting techniques. These types of techniques are very similar to those specified in chapter 2.1.2.1, as many of them offer means of selection and manipulation.

For an intuitive manipulation, one of the main components that needs to be present is the ability to change the object’s position, orientation and its center of rotation [Min95]. In order to make this possible, three types of inputs have been identified: hand-specified, physical controls and virtual controls.

[3]Google Cardboard design guidelines, https://designguidelines.withgoogle.com/cardboard/interactive-patterns/display-reticle.html, last access June 2019

Hand-specified controls are those in which the user uses his own hands to interact with the virtual world, creating an immersive experience. It is considered one of the most intuitive means available to change the position and orientation of a virtual object is to allow the user to "grab" it, and move it as though he were moving an object in the real world [Min95]. Among other variations, the grabbing mechanism can be done by holding a button pressed when the hand makes contact with the desired object. This action "attaches" the object to the hand, making it possible for the user to move around with that object. For dropping the object, the user simply has to let go of the button, and the object "disattaches" itself from the user's hand.

Physical controls are those that use some sort of external input devices such as a joystick, controller or a touchpad to control an object position and orientation. These types of controls are excellent for precise positioning of objects (since each degree of freedom can be controlled separately), but lack natural mappings and can make it difficult to place an object at some arbitrary position and orientation [Min95]. A more detailed explanation of these type of controls is present in section 2.1.1.3

Virtual controls are virtual objects or interfaces that are used to interact with other virtual objects [Min95]. A user can be provided with an interactive menu or toolbox and use it to trigger commands that can affect other objects. For example, a switch can be turned on to spawn a ball, or a virtual tennis racket can be picked up (using hand-specified controls) and then used to hit a virtual tennis ball.

2.1.2.3 Menu Interaction

Unlike 2D environments, menu placement in a virtual 3D environment must be carefully considered. The interaction should be "natural" and should stay away from the traditional WIMP (Windows, Icons, Menus, Pointer) methods [BWT01].

Kim et al. [KKP+00] indicates 3 types of placement for this kind of environments, also visible in figure 2.5:

- **World Fixed** (WF): The menu system resides at a fixed location in a "strategic" world location. This allows a relatively comprehensive display of the overall menu structure and menu selection history because it is located at a strategic location away from where the task is being carried out.
- **View Fixed** (VF): The menu system is attached at and viewed from a fixed offset from the user and moves along its tracked head movements. In this case, the dimension of the menu must be carefully considered, in order to not block the user's view.
- **Object Fixed** (OF): The menu system is attached to one or more virtual objects. Unlike World Fixed menus, which stay in the same place, this type of menus mimics the movement of the object it is attached to, whether it is translated, rotated or scaled.

Selection and manipulation with these menus can be done the same way it is done with common objects, with the restriction that the user cannot directly control their position and orientation.

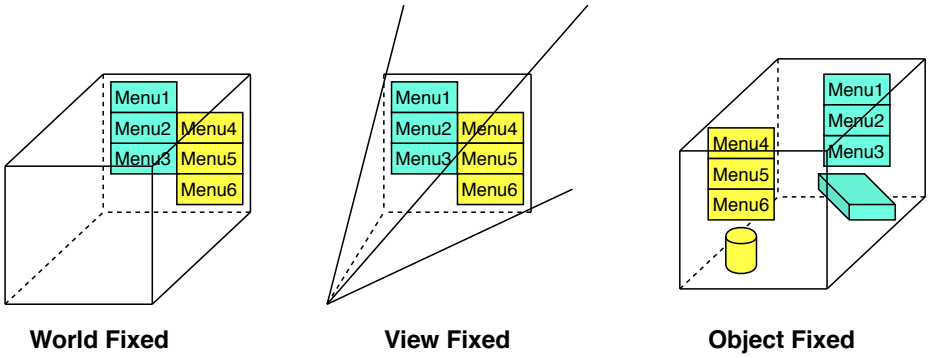


Figure 2.4: Variants of VR menu display, based of Kim et al. [KKP+00]



Figure 2.5: Menu interaction in the Oculus Quest

To interact with a menu, Mark Mine [MBJS97] refers that a *Look-at Menus* technique can be used to interact with them, in which the user gazes at the particular menu piece that he wants to interact with, using a press a button to trigger the activation. This type of interaction is also commonly done using laser pointer or touch techniques, specified in section 2.1.2.1, like in Figure 2.5.

2.1.3 Movement/Locomotion Techniques

Olivier et al. [OBKP18] concluded in their study that users had similar behaviors between moving in public spaces in real conditions vs virtual ones. This turns the act of moving in a virtual environment specially important to create an immersive VR experience, since the user is expecting some kind of movement he is already familiar within the real world. A plethora of techniques have been researched and improved, specially since the launch of the Oculus Rift Development Kit 1 in 2013 [BCK17]. Nevertheless, with all with their advances and qualities, there is still discussion about which technique should be universally used in terms of locomotion. Peck et al. [PFWM12]

defend that real walking is the most immersive locomotion technique available specially if combined with a Reorientation technique, useful when the virtual world is larger than the physical room the user is in. This allows the user to walk to the edge of its room, and then is forced to turn around to continue to walk in the same direction in the virtual world. But due to its physical limitations, Bozgeyikli et al. [BRKD16] suggest the point and teleport technique, which is related to be a much better approach in terms of VR sickness, but can break a bit of the immersion feeling for some users. This section aims to breakdown the most commonly used locomotion techniques, following the typology suggested by Boletsis [Bol17]. In his research, Boletsis suggests that locomotion techniques can be grouped into four different categories: motion-based, room scale-based, controller-based and teleportation-based.

Motion-based locomotion techniques are those that urges the user to have some kind of physical movement to enable interaction, while supporting a continuous motion in open VR spaces. These include:

- The **walking-in-place** technique is where the user simulates a walking motion, but stays stationary. This can be performed without support from external devices, or with some other kind of devices, like the VirtuSphere. The VirtuSphere [MFW08] consists of a large hollow sphere that sits on a special platform that allows the sphere to rotate in any direction as the user moves within in; it resembles a human-size hamster ball. This device allows the user to move in any direction in a virtual environment, without losing the sense of immersion, as they are actually walking inside the sphere, almost as if they were walking on a treadmill.

- **Redirected walking** is a technique that enables the user to walk on paths on the real world that vary from the paths they see in the virtual world, without perceiving this difference [LS18]. This is done by applying an unnoticeable mismatch between the user's real and virtual movements, forcing the user to reposition and/or reorient themselves in order to maintain their walking direction. An example of this technique can be visualized in figure 2.6.

- **Arm swinging** is a technique where the user swings his arms back and forth to generate momentum in the virtual world, causing the user to move in the direction they are facing. McCullough et al. [MXM+15] found on their experiences that their arm swinging method outperforms a simple joystick and that spatial orientation is comparable to physically walking on foot. Besides, this method does not suffer from space limitations and requires less physical energy than walking. The movements of the arm are inferred by the position of the controllers the user is holding, or by special armbands the user can wear, like the Myo armband [MXM+15]. The user can easily control the velocity of the movement, as it speeds up as he swings his arms more rapidly, and stops when the user stops the swinging motion.

- **Gesture-based** locomotion is when a user can move in the virtual environment using different gestures, or a combination of gestures, whether with their arms, hands, head and/or legs. These include taps, pushes [FPB+16], shaking the head or any combination of gestures that

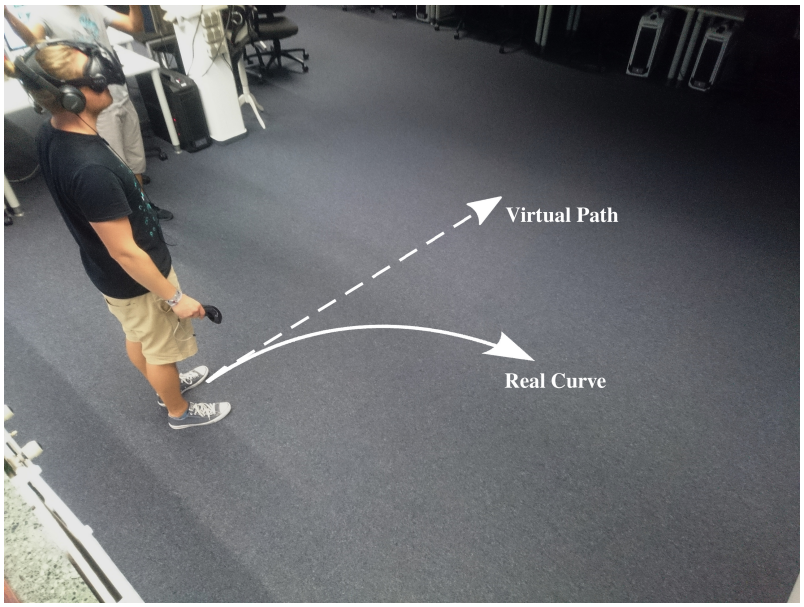


Figure 2.6: A user walking virtually straight while being redirected [LS18]

might be developed and can be tracked by devices such as the Leap Motion or the Microsoft Kinect [Bol17].

- **Reorientation** techniques are those that force the user to reorient themselves when they reach the limit of their physical space. Some variations of this kind of techniques are:
  - Rotating without warning, which consists on rotating the virtual environment when the user reaches the edge of the available space, without warning the user.
  - Rotating with audio warning. This variant sends an audio warning to the user that he is reaching the edge of the physical space, prompting him to turn around, rotating the virtual world to meet the new, correct orientation.
  - Distractor-based. Proposed by Peck et al. [PMFW09], a distractor can be defined as an object, sound, or combination of object and sound in the virtual environment that the user focuses on while the environment rotates, reducing perception of the rotation. Peck et al. [PMFW09] found that reorientation techniques based on distractors reduce the likelihood of a users' feeling as if they are turning around while being reoriented and also that subjects prefer reorientation methods with distractors and consider them to be more natural.

Room scale-based locomotion is done through **real-walking** techniques. It consists of users actually walking inside a physical room, replicating the movement on the virtual world. Literature states this as the most immersive locomotion method available [UAW+05, NSBK15, LS18, SVCL13, WKM16, PFWM12]. Usoh et al. [UAW+05] even cite that real walking "yielded a strikingly compelling virtual experience — the strongest we and most of our visitors have yet experienced". Wilson et al. [WKM16] support this conclusion, concluding in their research that

users make fewer mistakes in their trajectory if they use the real walking technique. But, with the grand limitation of being confined to the space the user has available in their room, it makes it difficult to traverse through a virtual environment if this environment is bigger in size than the user's physical room.

Controller-based locomotion uses controllers to move the user artificially in the VR environment, in a continuous motion. These include:

- **Joystick-based** techniques include any type of external joystick or controller to move the user through the virtual environment [Bol17]. These can range from game controllers, keyboards, mouses, and specific VR controllers such as the ones in section 2.1.1.3.

- **Human joystick** is a technique that consists in physically leaning or tilting as a means of translating the user forward in the virtual environment [HWJ14]. It uses special leaning boards like the Nintendo Wii Balance Board, which consists on devices similar to scales with pressure sensors on each side. As the user shifts its center of mass on a direction, the board also shifts, propelling the user to move in that direction in the virtual environment. This method, proposed by Harris et al. [HWJ14], claims to give the user a superior spatial awareness when compared to joystick-based techniques, and is on-par with walking-in-place techniques. virtual space

- **Chair-based** techniques are those in which the user sits on a chair that acts as an input device. The movements of the chair translate into the virtual world, including tilting and rotating. These techniques can have various implementations, including an everyday office chair, which has been shown to give better usability when compared to joystick-based techniques [Kit17]. It does, however, provoke a greater level of motion sickness when compared to other methods.

- **Head-directed** locomotion are techniques where the user head movements of the HMD control its movement. By tilting the head forward or backward, the user is moved in the virtual environment, and turns left or right by rotating the head. The motion speed can be controlled by the pitching of the user's head [Bol17]. Kitson et al. [Kit17] refers that in head-directed techniques users get a greater sense of usability with this technique versus chair-based and controller-based, but lacks in control and comfort. Also, a major limitation of this technique is that a user cannot move in a direction and look around at the same time.

Teleportation-based locomotion includes multiple variants of the **point and teleport** technique. It consists of pointing to the spot where the user wants to go, and then teleporting to that position, usually through the press of a button. Since this is a technique that does not involve any visible translational motion, motion sickness is reduced to a minimum, turning this technique into one of the most user-friendly techniques available [BRKD16]. Vlahovic et al. [VSSK18] has conducted research regarding this topic and supports this conclusion, showing evidence that users prefer the point and teleport technique in terms of comfort and overall quality of the VR

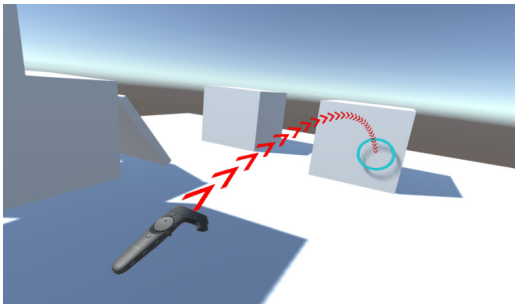


Figure 2.7: Point and teleport technique, implemented using the VRTK framework[4]

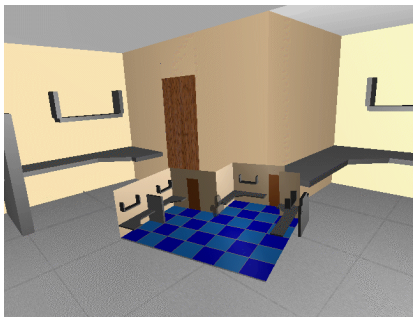


Figure 2.8: The world-in-miniature technique, showcase a miniature of the room the user is currently in [SCP13]

experience. Vlahovic also notes that users experience greatly increased motion sickness in other techniques when compared to this technique, while, on the other hand, concluded that it does slightly breaks the user immersion in the experience.

Boletsis and Cedergren [BC19] concluded that the walking-in-place technique gives a greater sense of immersion and enjoyment, but deteriorates if the usage of this locomotion mean is prolonged. Controller-based techniques also give a higher sense of immersion but the main issue is motion sickness. These types of techniques are usually well accepted among users, due to their familiarity with controllers from other types of applications like video games, regarding it as comfortable. On the other hand, teleportation-based techniques offer the least immersion feeling of this group of locomotion techniques due to its non-continuous motion, but are regarded as easy to use, and exceptional for fast traversal of a virtual environment.

Other techniques do exist, like the **World-In-Miniature** technique (figure 2.8), which allows the user to change his viewpoint through picking and relocating his representing icon in a virtual miniature replica of the virtual environment he is located in. Berger and Wolf [BW18] show that this technique outperforms controller-based and teleportation-based techniques in velocity of traversing a virtual environment, while also providing best spatial knowledge and causing the least motion sickness.

2.2 Computer-based Physics Simulation Visualizers

Physics laws explain/influence multiple of our sensations in our real world, some of them need to be considered to make a realist representation and some could be used as an advantage to represent a specific sensation, feeling or concept. These representations of physics principles can be applied for different purposes. But often, these physics concepts are very hard to represent in a real-world scenario, and even harder to visualize. In this line of thought, computer-based physics has become a must-have in order to represent these scenarios. In fact, it has been shown that

[4]VRTK Teleport Documentation, https://vrtoolkit.readme.io/docs/vrtk_basicteleport, last access June 2019

active engagement computer-based activities in learning processes are more effective than passive programs [DSZB05].

One of the examples of a physics-based application is the entertainment area; for this purpose the main goal is to use physics to create an experience/specific sensation. Although even for an entertainment purpose all the elements may respect the physics laws applied in that scenario. For example, the movement of an object that falls on the floor should be according to the gravity of all the environment, which may also include the representation of different gravity elements in the scenario to increase this sensation.

In this section, some examples of computer-based physics simulated will be provided, stating their main contributions to the field.

2.2.1 Dedicated software

There have been developed software dedicated to the sole purpose of recreating physics phenomena. An example of this is Algodoo[5], a free 2D physics sandbox from Algoryx Simulation AB, mainly designed for learning purposes. One of the key features of Algodoo is letting the user draw his own shapes and forms, and when finished, he can choose which type of object that is (whether it is a solid, a fluid, a gas, etc.) and it will react according to the physics laws in place. It also has the capability of taking a picture of a drawing in a paper or a whiteboard, and translate that drawing to the application. After the world is complete, the user can interact with it, whether by moving objects, cutting them or dropping new objects in the scene, and the world will react according to the physics laws in place.

The open-source project Open Source Physics[6] (OSP) is a more scientifically oriented software whose mission is to spread the use of open source code libraries that take care of a lot of the heavy lifting for physics: drawing and plotting, differential equation solvers, tools, and compiled simulations. Using this software, it is possible to implement different many different experiments in the Java programming language. An example of a project using this software is the Projectile Motion with Angry Birds[7] which uses the video game Angry Birds as a basis for a projectile motion problem and simulates it through OSP, as shown in figure 2.9.

Vortex[8] is a proprietary software consisting of a real-time physics engine, that integrates and tests mechanisms behaviours in simulated environments. It allows the creation of virtual prototypes, and simulate them in specific hardware with their own physics specifications such as operating a tower crane.

[5] Algodoo, Physics Game, http://www.algodoo.com, last access July 2019

[6] Open Source Physics, Physics simulator, https://www.compadre.org/osp/index.cfm, last access July 2019

[7] Projectile Motion with Angry Birds: https://www.compadre.org/osp/items/detail.cfm?ID=11562, last access July 2019

[8] Vortex, Physics engine, https://www.cm-labs.com/vortex-studio/, last access July 2019

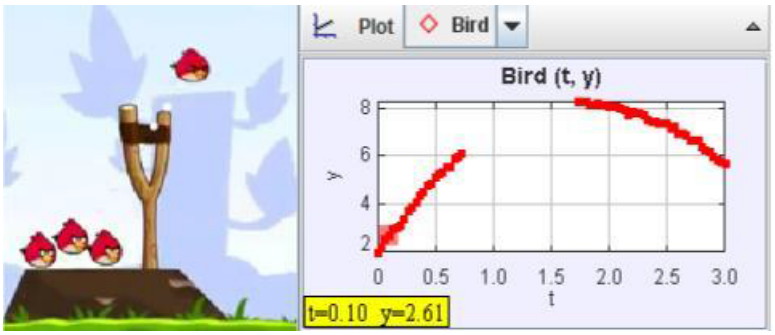


Figure 2.9: The Projectile Motion with Angry Birds project, using OSP[7]

2.2.2 Related software

All the cases reviewed until now had the singular purpose of providing a high-fidelity physics simulation environment. However, there are also applications that use some physics concepts but are not focused on these simulations, and aim to provide some other type of experience, using physics only as a way to complement and complete the experience.

The field of video games is a clear example of this, where many games employ some physics-based interaction along with the game experience. In this case, however, the physics laws that apply to a game, may not be similar to those of the real world, and objects may have different behaviours than what they have in the real world. Savage et al. [SMM+10] states that the first-person-shooter video game genre includes learning the physics of the simulated world through a process of experimentation. This suggests that this process of experimentation in a game-based application improves the learning capabilities of the player on physics subjects.

The Angry Birds[9] video game became widely popular due to its use of comical characters and also because of its gameplay heavily focused on physics. The objective of the player is to shoot multi-colored birds and hit the green pigs presented in the level, but since many of them are not in a direct line of sight of the birds, the player must aim and shoot at the structures present, either destroying them or causing them to collapse. After this, all the structures that were being held in place by the one destroyed will fall causing a sort of domino effect that is ruled by the physics laws of the game. Some instances of the game simulate physics laws different from those in Earth, one example being Angry Birds Space[10], done in collaboration with NASA, in which the player has to take into consideration the low gravitational pull of the projectiles, causing them to have a more straight-line trajectory instead of a parabolic one, as it occurs on Earth.

Additionally, in the cinema industry, movies like Interstellar[11] make use of computer-generated imagery (CGI) with particle physics systems to improve their scenes in ways it wouldn't be possible in a real-life situation. In the case of Interstellar, not only particle physics was employed, but also gravity and black hole simulations were conducted, in order to improve the realism of the movie.

[9] Angry Birds, Mobile game https://www.angrybirds.com, last access July 2019
[10] Angry Birds Space, Mobile Game https://www.angrybirds.com/games/angry-birds-space/, last access July 2019
[11] Interstellar, Movie, https://www.imdb.com/title/tt0816692/, last access July 2019

2.2.3 Game Engines

Game engines are software that provide numerous tools in order to develop applications (mostly video games) easily. Among these tools features rendering engines, animations, sound systems, but also, physics and collisions systems. These engines can work as physics simulators, specially in rigid body dynamics, but also in fluid and soft body dynamics.

Unity 3D[12] is one of these game engines, widely adopted in the video game community, with powerful support for modeling physical properties [MMSE17] and allows development mainly in C# programming language. It boasts a vast variety of options to modulate the physical world, such as indicating mass, velocity, drag forces and collision detection, among others. As an example, it is possible to assign a collider to an object that takes the exact shape of that object, or to assign it to completely different shape, one that it's not even touching the object itself, and will interact with the world only when that collider touches another collider, and not when the object itself touches it. Some projects using physics in Unity consist of the video game Kerbal Space Program[13], where players are asked to build a rocket to go to space, and recreates many of the physics interactions that occur in the rocket take-off, flight and in outer space; BattleTech[14], a giant-mech turn-based strategy game, in which laser and ballistics simulations are employed; many scientific research simulations are designed using Unity, whether for physics simulations [MMSE17] or for other purposes [NA19, KB18].

Unreal Engine[15] is a game engine similar to Unity3D concerning physics simulations with the difference that development in Unreal uses C++ programming language. Both engines have built-in physics engines and allow developers to further extend or even distort the physics capabilities of the engine. Price [Pri08] suggests that even though the Unreal Engine does have some caveats concerning actual physics representation, it is a viable solution and an important resource for studying many physical systems, for students and teachers alike.

2.3 Science Simulators and VR applications

Simulators are a way to provide experiences and to prove concepts that would be hard to reproduce in a real-world scenario. In particular, VR uses this approach in many areas, such as education or entertainment.

A VR application example for an entertainment purpose is flying as a VR experience. There are equipments designed to increase the representation and this experience include physics representation when that allows the simulation of moving wings, and to create a flying stability, as well as by including the possibility to position during the movement in such an angle that allows to increase or decrease the flying velocity.

[12] Unity 3D, Game engine, https://unity.com, last access July 2019
[13] Kerbal Space Program, Video game, https://www.kerbalspaceprogram.com, last access July 2019
[14] BattleTech, Video game, http://battletechgame.com, last access July 2019
[15] Unreal Engine, Game engine, https://www.unrealengine.com/, last access July 2019

VR scenarios are also being for learning and training in virtual environments. One example simulators of a real context as goes for an airplane pilot to practice his capacity to fly the machine in a virtual simulator, or the simulation of driving a car in a rainy weather, including the constraints of less friction between the car wheels and the pavement. For training, a VR scenario can allow to represent the application of huge force (which could be hard to represent in a real context) in order to understand the result of the application of this force in the other environment elements, the same goes for the representation of micro forces that occur in a not visible system allowing its understanding and to interact and see the impact analysis, which is also impossible to directly observe in the reality.

2.3.1 Medicine

Medicine is one of the fields that is taking advantage of technologies such as VR. This can be applied to computational neuroscience, molecular modeling, phobia treatment and ultrasound echographys [ADI17], for example. Also, and closely related to section 2.3.2 about Education and Training, medical simulators have been in place to help train new surgeons and medics, or for them to hone their skills. In fact, Seymour et al. [SGR+] shown that apprenticeship training with and without a virtual simulator, medics trained on a virtual laparoscopic cholecystectomy simulator completed the intervention in 29% less time and were 5 times less likely to injure the patient throughout the process. Andersen et al. [AS02] support these conclusions, having similar results in their study where surgical residents trained within a VR environment shown significant improvement in the procedures, averaging fewer errors per procedure and taking less time to complete it. NeuroTouch (see figure 2.10) is one of such simulators and allows simulating brain tumor removal using a craniotomy approach [DLDM12].



Figure 2.10: A neurosurgery resident training in the NeuroTouch prototype [DLDM12]

2.3.2 Education and Training

Those intimately involved with education generally agree that "experience is the best teacher." The reality of the educational process, however, is that students are seldom given opportunities for direct experience of that which is to be learned [LEB93]. Computer technology can help to fill this gap and provide ways to learn and interact closer to the reality than books or other conventional educational means provide. With the advent of internet, many e-learning platforms arose, like Udemy[16] or Udacity[17] and provide greater opportunities for people to learn or to perfect subjects that school couldn't provide them. Many of the courses provided in these e-learning platforms do provide a significant number of exercises and guide their students through them in order to ease the learning process. But even though this can be perceived as a way to more easily obtain knowledge, it still lacks the fundamental direct interaction, which is essential in some subjects.

Some advances to provide VR learning platforms have been made. In 1993, Loftin et al. [LEB93] developed a Virtual Physics Laboratory in order to help students to have a better grasp of common physics misconceptions, like the "motion implies force" notion. Nelson and Ahn [NA19] have also developed a VR application at the Iowa State University to teach students professional development skills using game-based learning frameworks. Because of the greater sense of immersion and presence that a VR application delivers, and the ability to provide its users with improved interaction means over traditional technology like a desktop computer with mouse and keyboard, it has been theorized that students do get a better understanding of subjects that are taught through VR interfaces. In fact, in a study conducted at The Australian National University and The University of Queensland, Savage et al. [SMM+10] claim that students who had a practical lesson in a specific topic (Real time Relativity) within a VR setting not only affirmed to better understand the subject, but they did score on average 0.8 points (out of 10) in their final exam about the subject compared to those students that had traditional lectures. This immersive and engaging system has lead to additional research into virtual reality as a teaching tool because it can be a strong motivator for the student to continue engaging in the learning process [NA19].

2.3.3 Entertainment and Culture

Video games are one of the areas where VR has had a wider adoption. In fact, Steam reports that from 2016 until 2018, the percentage of users playing in their platform with a VR headset doubled. And while, by the end of 2018, the total number of users playing with a VR headset was of 0.8% as presented in figure 2.11, it has increased to 1.0% as of June 2019. The most played VR video game on Steam (Beat Saber[18]) as of July 2019, has currently roughly 1500 players per hour.

The use of VR may also be applied for social purposes, as is the example of a project, created at the North Carolina State University called Distance Education and Learning Technology Applications (DELTA), where VR is used to create a virtual scenario where the students can experience

[16] Udemy, e-learning platform, https://www.udemy.com, last access July 2019
[17] Udacity, e-learning platform, https://eu.udacity.com, last access July 2019
[18] Beat Saber, Video game, https://store.steampowered.com/app/620980/Beat_Saber/, last access June 2020
[19] Steam usage data taken from the Steam Hardware & Software Survey: https://store.steampowered.com/hwsurvey

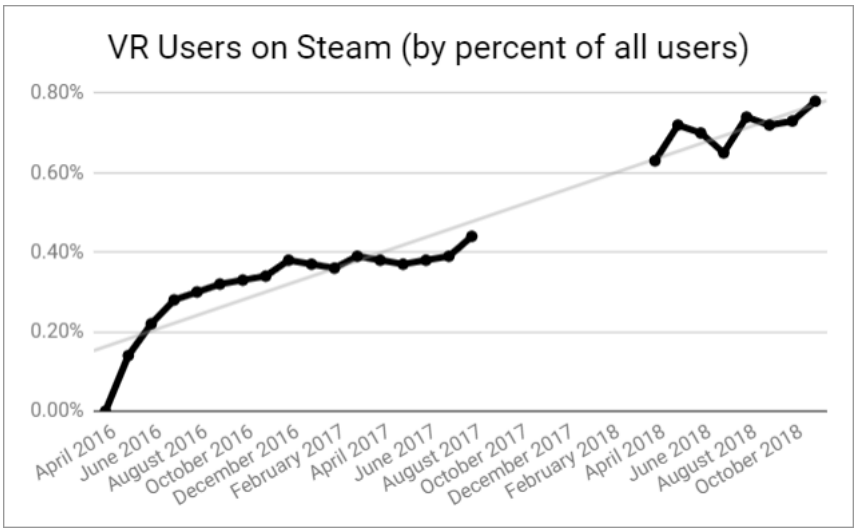


Figure 2.11: Steam total users playing with a VR headset. (the seven months gap is due to erroneous data from a Valve collection issue)[19]

the view of the same situation but being a different person from a different culture for the story, in order to visualize the impact of cultural differences on people's lives. They state that "the resulting experience gives students a sense of visual and audible presence that differs from traditional media forms."[20]

VR is also being applied in cultural reconstruction, where the representation could be done by using 3D heritage applications that can help the user to examine the architectural details. Nevertheless, it still does not allow the user to have an idea about how the site has been enacted in the past [BNS09]. According to Bogdanovych et al. [BNS09], creating 3D virtual worlds is being credited as the most affordable, dynamic and interactive option for integrating the environment, the artefacts and the knowledge associated with a culture. Another example of this cultural recreation is the British Museum who a "Virtual Reality Weekend" event [RE16] in 2015, allowing users to explore a scene during the Bronze Age where they could interact with an object and hear a description about it.

2.3.4 Engineering

Before the use of VR in the engineering field, descriptions of engineering components were viewed as lifeless drawings or static perspective projections some of which were animated along a set path through a 3D model. VR has enabled components to be virtually manufactured, inspected, assembled and tested without the need for expensive and time consuming prototype production [Cox03]. Portman et al. [PNFG15] cite the use of VR in engineering and architectural fields as a technology that better supports the decision-making process, and that it helps moving from conceptual ideas to more concrete knowledge. It is pointed, however, specially in the field of

[20]Information taken from: https://campustechnology.com/articles/2018/08/21/using-vr-to-help-students-understand-cultural-differences.aspx

environmental and landscape architecture, that behavioral and ethical challenges may arise as the use of VR may trigger an increased behavioral change and environmental protection as these relate to climate change adaption and response.

In the automotive industry, VR has hit a point where is now essential as now designers and engineers work on one and same digital model, fully utilizing the model rather than building several prototypes [FAR15]. Additionally, automotive companies now also provide Virtual Test Drives like the Volvo XC90 (see figure 2.12) where potential car buyers don't need to go to showrooms to test drive a car, and can compare car models of different brands without having to leave home by simply installing an app on their phone, and use it on a mobile VR headset like Google Cardboard. Not only for the customers, but these Virtual Test Drives are also more convenient for car retailers as they don't need to set up store with the physical car displayed [FAR15].

VR has been applied in the architecture field, in which the difficult creation of prototypes and models provides a notion of reality more distant than what happens in a prototype of a software. In order to ease this process, applications like Eyecad VR[21] are being developed. This application allows the architects and engineers to quickly assemble a virtual 3D environment from a preset of assets (or importing their own) and create interactive objects which can then be manipulated and traversed through using a HMD. This way, architects can use VR to take themselves, or their clients for a walk throughout the rooms of buildings they are designing, allowing possible design changes to be visualized. Another advantage of these VR walkthroughs instead of the common CG animations is that the viewer is not restricted to a preset path, and they can explore the virtual environment freely [ADI17]. Kunz et al. [KZFN16] designed an application where people can use real walking and redirected walking techniques (see section 2.1.3) to explore a virtual factory. They stated that this method has become part of the design process of the factory as it allows for a realistic approach to what the factory would be in real life, and it engages everyone who participates in the experience in the design process.

[21] Eyecad VR, VR Architectural software, https://www.eyecadvr.com/pt/, last access July 2019

[22] Volvo XC90 Experience: https://www.volvocars.com/us/about/our-points-of-pride/google-cardboard, last access July 2019



Figure 2.12: Volvo XC90 Virtual Test Drive[22]

2.4 Summary

In this chapter, it was seen how presence and immersion take fundamental roles whether in choosing which equipment users should use, and also how it affects interaction with the virtual world. Specifically, locomotion is a major step stone in the way for a fully immersive and captivating VR experience, and the adaptation of the locomotion means to make the user comfortable and engaged are crucial. Table 2.2 shows an overview of the different techniques identified, as well as in which category and sub-category do they fall.

Table 2.2: VR Interaction Techniques Overview

Technique	**Category**	**Sub-Category**
Touching	Selection/Manipulation	Local
Grabbing	Selection/Manipulation	Local
Go-go	Selection/Manipulation	Local
Laser Pointer	Selection	At-a-distance
Spotlight	Selection	At-a-distance
Occlusion selection	Selection	At-a-distance
Gaze directed	Selection	Gaze directed
Voice Input	Selection	Voice Input
Laser Pointer	Selection	At-a-distance
World Fixed	Menu Interaction	-
View Fixed	Menu Interaction	-
Object Fixed	Menu Interaction	-
Walking-in-place	Locomotion	Motion-based
Redirected walking	Locomotion	Motion-based
Arm swinging	Locomotion	Motion-based
Gesture-based	Locomotion	Motion-based
Reorientation	Locomotion	Motion-based
Real-walking	Locomotion	Room scale-based
Controller/Joystick-based	Locomotion	Controller-based
Human joystick	Locomotion	Controller-based
Chair-based	Locomotion	Controller-based
Head-directed	Locomotion	Controller-based
Point and teleport	Locomotion	Teleportation-based

In addition, an overview of how physics are being simulated on a computer basis, and where VR has been applied was also an approached subject.

In the following section, the proposed framework will be defined, with a focus on customization, and aims to create a test-bed in which the techniques researched in this section can be implemented and thoroughly tested.

Chapter 3

Requirements and Architecture

Having reviewed the relevant state of the art concerning VR interactions as well as physics simulations and its applications, this chapter presents the planning phase of the development of the project. It starts with a general analysis of the problem in section 3.1, followed by the system requirements that were specified for this solution, which are detailed in section 3.2. The overall system architecture is described in general in section 3.3. Lastly, the usability principles that were considered for this project are detailed in section 3.4.

3.1 Problem Analysis

Physics simulators are useful to represent concepts that are hard to see in a real-life situation, as was debated in section 2.2. VR expanded these simulators boundaries to facilitate this type of simulations, allowing new ways of interaction that desktop simulations could not afford.

Some of these concepts can be recreated using common objects. For instance, the representation of the movement of a basket ball can help to understand the associated physics concepts, such as gravity, velocity, acceleration and drag caused by air resistance. Recreating a projectile expected trajectory while being able to manipulate it in real time can help to get a more practical hands-on approach when compared to traditional educational methods.

Another key factor of this dissertation is the VR interaction paradigm, as users must use these interaction techniques in a way that feels natural and realistic, even if using objects or are facing situations that would be impossible in real-life. As such, these interaction techniques must be carefully considered, or these can become a hindrance towards the objective, degrading the overall experience. These techniques must become a tool for reaching an objective, one that can be quickly learned and used without much thought, even if they seem complex at first glance.

Combining these aspects with a collaborative environment, where users must join their efforts to reach a common goal can also foment knowledge exchange, while also turning the experience in a fun, more engaging one.

3.2 Requirements

Considering the scenario with the best result in the previous section, a list of requirements was needed in order to organize and prioritize the features and components to be integrated. These requirements were divided under 2 groups: system, interaction and collaboration.

Under the interaction requirements, the following were created:

- **Remote interaction** - The user should interact with an object remotely, without actually touching it in the virtual world.
- **Data manipulation** - The user should be able to change parameters of an object, like velocity or direction, in a 3 DoF context.
- **Data visualization** - The user should see in a clear way the values of the parameters he is changing, along with its consequences.
- **Time control** - The user should be able to stop and resume time at his will, stopping the movement of all desired objects, and then resuming its movement.

The application should have a multiplayer component, and thus, the following requirements concerning this aspect were thought of:

- **Online Collaborative Environment** - The application should be able to accept at least 2 users simultaneously, each with its own HMD. A desktop mode is not required, but cross-compatibility would be a good addition.
- **Collaborative** - The application should encourage collaborative behaviour between the users towards a common objective
- **User visualization** - Each user should see other users in the scene and their actions.
- **Asymmetrical mechanics** - Each user present in the scene should have different ways of interaction, making it difficult to reach an objective alone, but easier together.
- **Non-verbal communication** - The users must have some way of communicating with each other inside the application, without relying solely on verbal communication in the same physical room, or on external video or audio tools.
- **External settings** - At the start of the application, it should be able to read an external settings file (such as a .json file) and interpret its content. Changing the file should trigger different behaviours on the application. This allows for parametrizing an experiment or scenario without code changes.

Some additional requirements were defined for the system, that although not required for this work, would enrich it and contribute to a more robust application:

- **Device compatibility** - The application should run on the Oculus Rift, using its motion controllers, but running on more systems is recommended.

- **Save usage statistics** - The application should save the most relevant usage statistics for later interpretation.

- **Minimum FPS** - The application should run with a minimum FPS equal to the HMD's native refresh rate, to ensure a smooth experience and minimize motion sickness.

3.3 System Architecture

The purpose of this section is to display the defined architecture of the application, as well as to explain the decisions behind it.

The planned architecture can be observed in Figure 3.1. Each user will use their HMD and controllers as an input to interact with the scenario. This scenario will have different parameters which can be set via an external JSON file, which is interpreted prior its instantiation. These world parameters are global settings for the scene, that do not change the overall experience, but add slight changes to the environment, augmenting the level of customization on what the user will interact each time he runs the program. Examples of these parameters are the world gravity or object's weight and drag. The first user to join a room will become a Master Client. The Master Client is responsible for the initial configuration of the room, connection to a remote server and *"owning"* the necessary objects, meaning that only he can control and manipulate said objects. All communications with the server pass through a Network Manager, which interprets, parses and delivers both sent and received communications. A second user that tries to connect to the server will firstly verify if there is already any room created in the same locale, and proceeds to join it if this verification turns positive. This second user becomes the Host Client. After both users are connected, each instance will run independently from the other, but some objects need to be synced between instances (e.g.: the user's avatars, common objects). This synchronization process differs slightly in both clients. The objects who are owned by a client are send over the network to the other, so they can be replicated in the host client. Also, the Host Client will accept any change that comes from the Master Client, but the Master Client will filter incoming communications and only accept certain changes. This aspect will be covered in detail in section 4.6. Finally, during the application, some usage data is being captured for later analysis, which, before the application is closed, is saved in the client's local storage. Although some of this data is common to both clients, some other is not, and as there is no need to implement a mechanism for exchanging all usage data between all application instances, both should be retrieved and interpreted in a single play session.

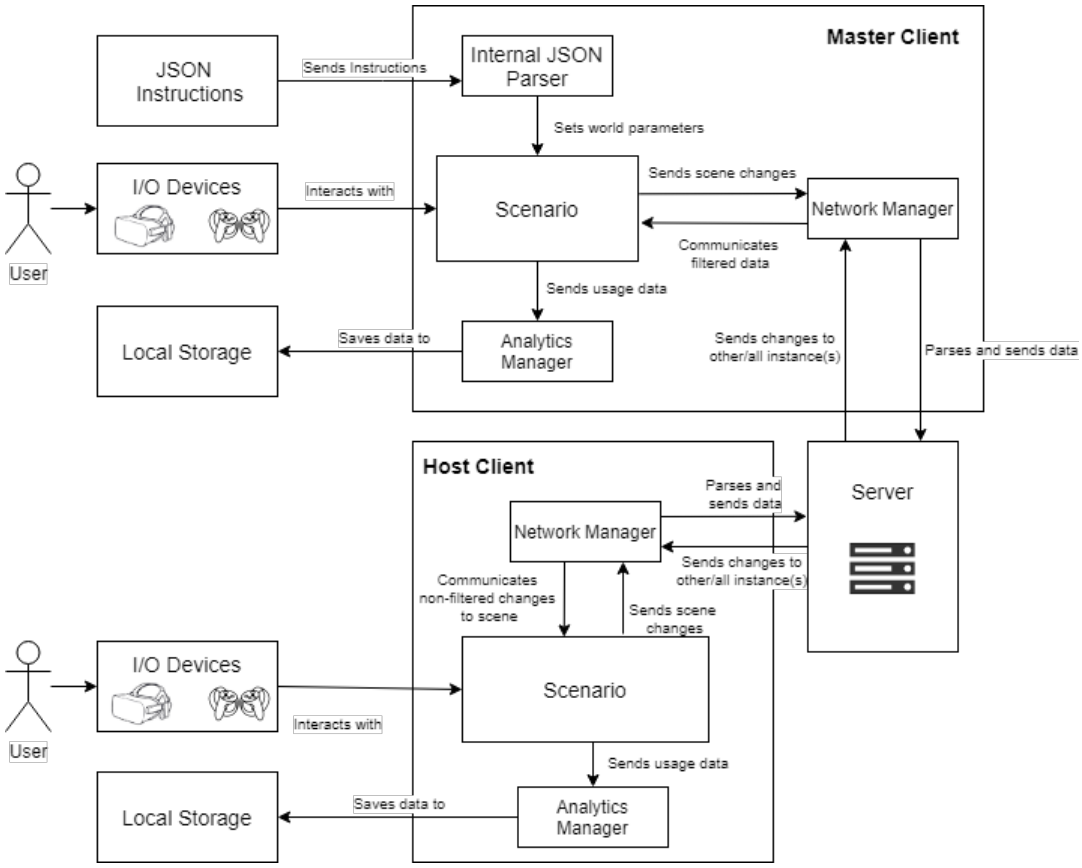


Figure 3.1: General System Architecture

3.4 Usability

Before starting with the implementation of the project, it's important to establish some fundamental bases regarding usability, either to ease the users into a VR world or to introduce them to a new interaction paradigm that they might not be accustomed to.

The experience should be designed to be interacted with in a standing position, allowing for a full 360º rotation and traversal through the virtual world. A HMD will be connected to a computer and the user will use the motion controllers available to interact with the world.

One of the key aspects of this dissertation is to provide the users with methods of locomotion and interaction with virtual objects, without compromising the immersion.

Regarding locomotion, creating an interactive menu for the user to choose his preferred method is out of the scope of this project, but in order to accommodate as many users as possible, and minimize possible effects of VR motion sickness, at least two of the locomotion techniques researched in section 2.1.3 should be implemented.

Interaction with the world's objects remotely is a critical component in this work as it must allow the user to move and rotate them in any direction, and even throw them with a specified velocity, mimicking hand-specified object manipulation studied in section 2.1.2.2. Taking this into

consideration, a representation of the virtual object should be created, allowing the user to manipulate only this representation and not the object itself. Some visual feedback must be implemented to allow the user to understand that the manipulation done into the representation will follow to the virtual object, allowing an indirect manipulation and creating expectations regarding the results. Another aspect regarding interaction, is the visualization of the effect that the current manipulation will actually have on the original object. As an example, when using a hand-specified technique to throw an object, even though the user doesn't see the exact outcome of the interaction, he does know that the object will leave the hand in the direction he points it with his hand. The exact position of where the object will fall cannot be pinpointed without other type of feedback, but the user does have some control over the area where it will fall, as it happens with the same movement in a real world situation. This exact behaviour needs to occur in a remote manipulation setting, where a user must know what will happen to the object, as this sensibility over the manipulation of the object is expected to decrease using this technique. As such, some additional feedback, either visual, auditory or haptic must be added to the interaction to counter this loss of sensibility on the interaction.

Concerning the multiplayer aspect, the main aspect for this experience is focused on collaboration between the users. The first main component for this aspect is to allow both users to see each other, giving each of them spatial awareness of the other, allowing for a greater sense of presence. In order to do this, an avatar should be created for each user, which, ideally, should be different from each other if there are more than two users in the same experience, allowing for easy identification of all parties. The next desired component for this aspect is giving the users a common goal to achieve, something that rewards collaboration between the users inside the virtual world, and discourages solo interaction. The final component desired for the multiplayer aspect is an implementation of asymmetrical mechanics. The users can have some common aspects, like the available techniques of locomotion, but they should be given different means to interact with objects, in a way that they complement each other, making achieving their common goal easier compared to them doing it alone. As such, and because these users may not be in the same physical room, some method of communication between the users is needed to overcome this obstacle, without relying, but possibly complementing external applications like communication systems such as Teamspeak[1] or video conferencing and chatting systems such as Skype[2] or Discord[3].

An additional aspect for this project is the possibility of studying how users can adapt to different circumstances and conditions, using the same mechanics and know-how they previously acquired. As such, the inclusion of an external settings file is desired, which can be easily modified without need to modify the internal code of the project, and is interpreted before running the application. These settings will not be able to change the whole scenario, but do change some aspects of it, such as values of gravity, air resistance or even include additional forces in the scene, increasing the challenge for the users and forcing an adaption to the new conditions.

[1] Teamspeak, https://www.teamspeak.com/en/, last access June 2020
[2] Skype, https://www.skype.com/en/, last access June 2020
[3] Discord, https://discord.com/, last access June 2020

As stated in the interaction requirements in section 3.2, one of the desired inclusions is a system of time control, meaning that the user must be able to stop and resume time at will. This mechanic must be carefully considered, as the user may not want to stop the movement of all the objects present in the virtual world, but only to control the movement of one specific object or even a group of objects. This mechanic should be simple to use and should not interfere with the ability of interacting with the stopped objects nor limit the freedom of moving around. It should also guarantee the possibility of resuming time with the world in the exact state it was before stopping it. Additionally, a method to not only stop and resume, but to slow down or speed up the movement of desired objects should be considered as it allows for even more control over the parameters of the objects involved.

Finally, some method to evaluate the usage of the application is needed. Requesting an user to achieve an objective with this technique multiple times while keeping track of every interaction the user has will help to understand the level of efficiency of this technique. In order to see how time control can help the user to reach his goal, every time the user uses this mechanic should be saved, along with the present conditions when he used it. This will allow to understand not only if this time control is effective in helping to reach a goal, but under which specific conditions it is used and what motivates the user to use it. Regarding the multiplayer aspect, it will also be a point of interest to understand whether the cooperative mechanics will actually help the users to achieve their goal, or in contrary, they hinder the users' performance. With this in mind, the communication attempts each user makes should be saved along with the events unfolded by that communication.

All of this data should be saved without compromising the application performance and without the user actually knowing that data is being collected in real-time (nevertheless, they must know that will happen prior to their usage).

3.5 Summary

In this chapter, an overview of the problem at hand was detailed, along with the deciding factors of this work. Taking this into account, a list of necessary requirements was introduced, divided into three categories: interaction, multiplayer and system. An architecture was proposed for the system, detailing how the external settings can be interpreted, how the communication between clients will be handled and also the functionality of an analytics manager, that will allow to get the usage data of the application. Finally, the usability aspects concerning this application were discussed, detailed in the various topics that concerns this work that will culminate in the implementation of the system described in chapter 4.

Chapter 4

Implementation

This chapter will present the implementation details of the prototype developed for this dissertation. Initially, a description of the case study chosen to depict the wished experiment is presented, coupled with the reasons why this case study was selected in section 4.1. The tools and technologies used along with the 3D assets that were obtained for this project are detailed in section 4.2. The resulting interfaces and visual aspects of the project are presented in section 4.3, while the interaction/locomotion aspects can be seen in section 4.4. These interaction aspects are tightly coupled with the physics simulation mechanisms referred in section 4.5. Other important component of this dissertation is the multiplayer, collaboration and networking aspects, which can be found detailed in section 4.6. The inner workings of the usage of external settings and how data is retrieved from the application can be found in sections 4.7 and 4.8, respectively. Finally, a summary of this chapter is found in section 4.9.

4.1 Case Study

Taking into account the requirements specified in section 3.2 and all the usability aspects discussed in 3.4, and to ensure a reliable experimental setting, five scenarios were considered as case study candidates.

1. **The Planetarium** - In this scenario, the user had the objective to understand the behaviour of gravitational attraction of celestial bodies. As such, he would send a satellite from a planet to orbit around a specific moon. There would be a difficulty progression system, in which at first, at planet Earth with just one Moon, the satellite launch would be easier as there are fewer bodies involved. Next, after launching on Mars, with 2 moons, the gravitational field of one would influence the other. At Jupiter, as there are 4 moons, the objective would be even harder.

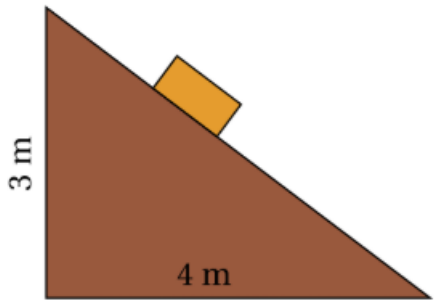


Figure 4.1: Physics problem, requesting to calculate the friction force on the sliding block. [Vil19]

2. **Roller slideway ship launching** - One of the last phases of ship building consists on launching the ship to the sea. Although various techniques exist for this phase, one of the oldest being used is the roller slideway launching, consisting in building a ramp (usually of steel or wood) and letting the ship slide down this ramp with the force of gravity [OC14]. This ramp would have to obey different parameters for the launch to be successful, as the ramp inclination can't be neither too steep as it may damage the ship on the impact with the water, or too gentle, as the ship may not move at all. This ramp would have to be customized for each individual ship, depending on factors like the material of the ship and its weight. In a simplified way, this is a common physics problem in engineering courses, like the one in Figure 4.1. It would be possible to build a simulation for these ship launches, in which the user needed to change different parameters to build the adequate ramp for each ship.

3. **Physics Laboratory** - This scenario would be used to study the behaviour of inanimate objects. The user could throw an object, with a fixed weight, to hit a target or to hit a certain objective (eg.: hit a target height X; make the object go as far as possible). It would be possible to see the values of velocity, acceleration and weight at any moment through a mechanism to stop movement of all objects in the exact position and to make them stay where they are the moment this mechanism is used.

4. **Gravity car race** - A gravity car is a motorless vehicle which is raced on a downhill road, propelled by the force of gravity. This scenario would consist of 2 players competing head-to-head to see which one would build the better and faster car. The players would be able to build the car using different pieces for the wheels, chassis, bumpers, etc, where each piece would give different attributes to the car. The cars would then race without direct interaction from the player, but as they break down during the race, the player would need to verify the car and repair it on the fly. A possible way for the player to move quickly to the location would be by using the WIM technique, verified in Chapter 2.1.3.

5. **Bridge Building** - This scenario consists of the user building a bridge over a river, by setting a number of anchor points and pillars as can be seen in Figure 4.2. In this scenario, two users

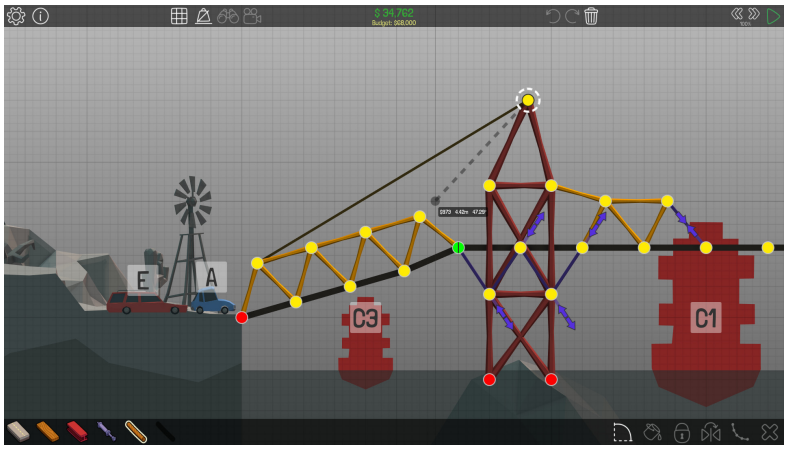


Figure 4.2: Various anchor points defined in the video game Poly Bridge.[1]

could begin building a bridge from each side of the river, until they meet in the center, and then see the action unfold.

In order to select which one would be developed as of this work, there was a need for some parameters to be defined, in order to assert the relevance for the project. The following parameters for the case study evaluation were defined.

- **Param. A** - Based on a difficult to simulate real-life situation
- **Param. B** - A physics problem
- **Param. C** - That would provide a setting to explore multiple ways of interaction and locomotion in VR
- **Param. D** - Collaborative or competitive
- **Param. E** - Immersive experience, that would justify the use of VR versus traditional simulations

Based on these parameters, all the scenarios were scored on a scale of 1 to 5, where 5 is the higher grade. This would help determine which of the 5 scenarios would be a better suit for the case study.

Based on these results, scenario 3, consisting of a physics laboratory, was chosen to be developed in this project. After a number of iterations on more practical aspects of the scenario, the final scene will consist of a basketball game simulation, in which the user will indirectly control the ball and try to score points to hit certain objectives. The user will control the ball remotely, using a proxy-based interaction paradigm, in which he will have a copy of the ball attached to his hand, and manipulate a vector to dictate the velocity and direction of the throw. This copy, similar to a hologram of the ball itself, will be the sole point of communication between the user and the

[1] Poly Bridge, Video game, https://store.steampowered.com/app/367450/Poly_Bridge/, last access June 2020

Table 4.1: Scoring all scenarios on the defined parameters

Scenario	Param. A	Param. B	Param. C	Param. D	Param. E	Total
1 - Planetarium	4	4	2	1	2	13
2 - Ship Launching	3	4	3	2	2	14
3 - Laboratory	5	5	3	3	4	20
4 - Car Race	2	3	5	5	4	19
5 - Bridge Building	5	5	2	3	3	18

ball, as the user should not interact directly with the ball, either by grabbing it or touching it. By manipulating it remotely, it should become easier to see where the ball is heading, based on the vector of the hologram.

4.2 Tools and Technologies

In an early stage of the planning phase, it was decided which tools and technologies to use in the development of the prototype. This work is derived from the master thesis created by Pereira [Per19] , which already had a set of tools used. Contrary to said master thesis, this one has a greater impact on physics components and interaction, however, since a ground basis was already established, it was decided to not derive from the previous work, and to reuse as much content as possible from it. Regarding this, Unity[2] was chosen as the game engine to use, since it already incorporates a Newtonian physics simulation environment, which could be customized to the needs of this project. As such, C# was the programming language used to develop the prototype, as it is the core language supported by Unity. Regarding the networking components of this project, in order to facilitate interaction and synchronization of needed objects between different instances of the game, Photon PUN[3] was the service chosen, as it offers a free account with limited usage. Also, slight changes were necessary to some of the 3D objects that were used in this project, and as such, Autodesk Maya[4] was chosen. Blender[5] was also considered for this work, but Maya was used as some of the assets used were in a native Maya file format, and even though it uses a subscription service, it offers a free month trial which was enough for the work that was needed.

In order to give some visual fidelity to the experience, the following free visual assets were also used:

- Basketball model[6], to represent the ball
- A hologram shader[7], to give the wanted representation of the ball a hologram feel. This shader was lightly modified according to the project's needs.

[2]Unity Engine, https://unity.com/, last access June 2020
[3]Photon PUN service, https://www.photonengine.com/en-US/PUN, last access June 2020
[4]Autodesk Maya, https://www.autodesk.com/products/maya/, last access June 2020
[5]Blender, https://www.blender.org/, last access June 2020
[6]Downloaded from https://www.turbosquid.com/FullPreview/Index.cfm/ID/371863, last access June 2020
[7]Downloaded from https://github.com/andydbc/HologramShader, last access June 2020

- Free Primitives[8], to use some primitive types not native to Unity, such as a cone for the vectors' arrows.
- Cursed Timer Font[9], for the scoreboard's text.
- Billboard[10] for the scoreboard itself.
- Basketball Court[11] for the court. The actual basketball board and hoop were removed for this project, as they didn't meet the level of detail wanted for the experiment.
- Basketball Board[12] for the basketball board and hoop.
- Trees[13] for the trees to help represent the wind.

4.3 Project interface

The chosen scenario to be represented in section 4.1 indicates that a basketball game scene must be recreated. As such, a minimal scene was built, where the user can walk around a basketball court, having a basketball board on one of its ends, standing some meters above him. The objective of the user is to score as much baskets as possible under certain circumstances, fulfilling the missions that are given to him.

This experience was built to be used in as many HMDs as possible, and it was tested successfully with the Oculus Rift and Oculus Rift S. For the testing phases, the Oculus Rift S was the used one, along with its controllers. However, due to some limitations on equipment during development, a desktop mode was also implemented, although not with all functionalities. This desktop mode allows only to traverse through the virtual world, and to communicate with the other user in a multiplayer setting. It is intended only for development purposes, as there are key components (like the hologram manipulation) that are missing.

When the application starts, the system verifies if there is an HMD attached, and if it is, it loads up all necessary VR components and spawns the controllers in the form of hands. If there is no HMD attached, the system will automatically load up the desktop mode. Whether it loads up the VR or desktop mode, the system will also spawn the user's avatar, which himself cannot see, as the whole experience will run in a first-person view perspective. This will be useful for the multiplayer component, where users can see each other.

[8] Downloaded from https://forum.unity.com/threads/release-free-primitives.234595/, last access June 2020
[9] Downloaded from https://www.fontspace.com/cursed-timer-ulil-font-f29411, last access June 2020
[10] Downloaded from https://assetstore.unity.com/packages/3d/environments/urban/billboard-9700, last access June 2020
[11] Downloaded from https://3dlancer.net/en/freemodels/sports-equipment/3dmodel-basketball-court-12931, last access June 2020
[12] Downloaded from https://free3d.com/3d-model/basketball-hoop-47554.html, last access June 2020
[13] Downloaded from https://assetstore.unity.com/packages/3d/vegetation/trees/realistic-tree-9-rainbow-tree-54622, last access June 2020

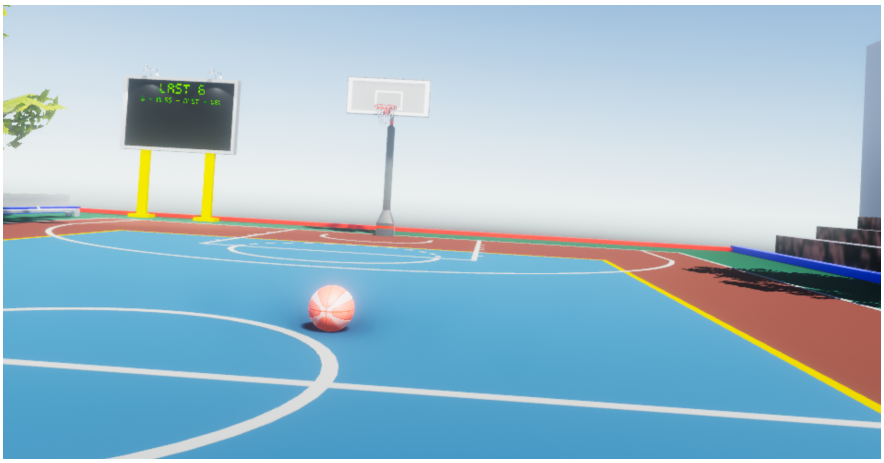


Figure 4.3: Basketball court, board and ball.

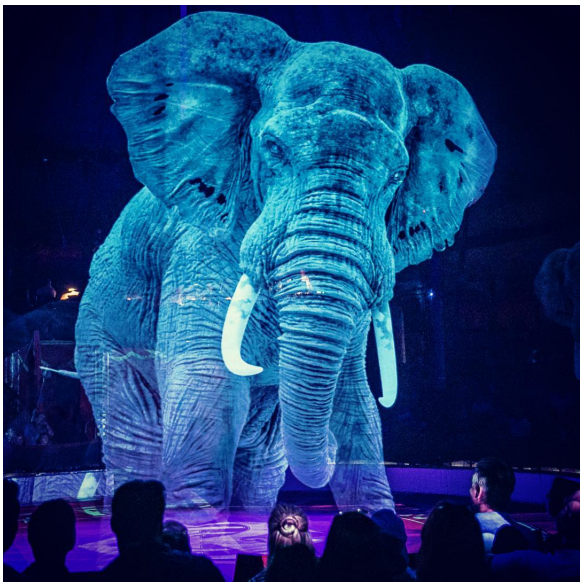


Figure 4.4: A hologram to represent an elephant in Circus Roncalli.[14]

The user is spawned in the center of the basketball court, and a ball spawns right next to it, as can be seen in Figure 4.3. From this point, the user can move freely using the locomotion methods listed in section 4.4, and is limited only by the boundaries of the court.

Creating a way to interact with the ball indirectly was one of the most challenging factors in this work. It needed to be a way that was both engaging, easy to use and that would give a substantial advantage over a direct manipulation of an object. A **hologram** is currently used in the industry as a way to represent objects or figures that would not be possible to physically see them. An example of this is the usage of holograms in the circus, as seen in Figure 4.4, where due to the ban of live animals in circuses, holograms have been used by some companies to replace these animals, effectively creating a representation of these in a grander scale.

[14]Image from Circus Roncalli Facebook page, at https://www.facebook.com/CircusRoncalli, last access June 2020

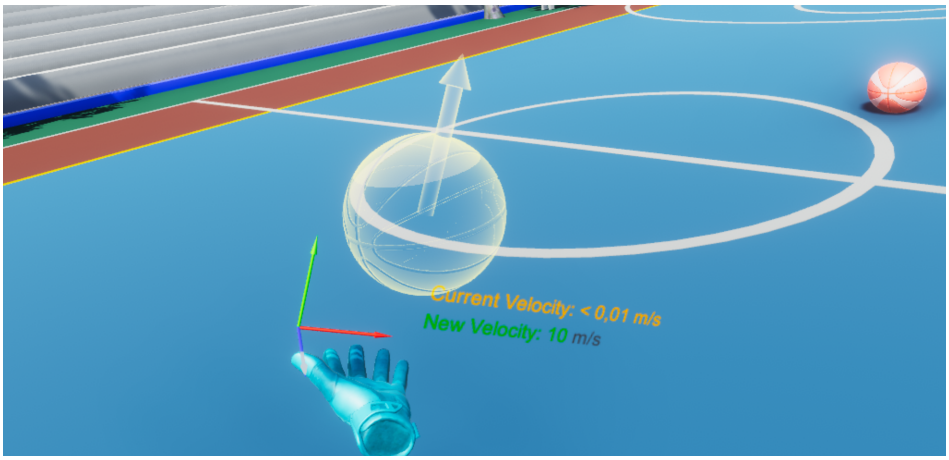


Figure 4.5: Hologram of the original ball.

Taking into account the advantages of VR interactions, where one can create different ways to interact and manipulate objects, different from what happens in the real world, it was thought that creating a hologram of the object it was intended to represent and manipulating this hologram could be an interactive way to remotely control the original object with a greater precision, when compared to direct manipulation.

The user can spawn the hologram in their left hand by pressing the "X" button on their left controller. The hologram then pops up, being a representation of the ball that is on the field. This representation includes a different shader to give the impression of looking like a hologram, with some configurable parameters, such as line intensity. The hologram will copy the rotational movement of the original ball, to give some visual feedback that moving one object will affect the other. This behaviour can be seen in Figure 4.5. Along with the hologram ball, an arrow comes out of the center of the hologram, representing the velocity vector that the user will apply on the ball. How these controls can be managed will be detailed in sections 4.4 and 4.5. The user can then grab the tip of the vector to manipulate it. For this, the user must get his right hand close to the arrow, and it will glow green, indicating that it is in position to be grabbed. After grabbing it, using the lateral grab button on the controller, the vector becomes attached to the right hand, where it can be moved freely. This movement is divided into two sections: rotation, where the user can rotate the vector around the hologram's ball, affecting the movement's direction; and scaling, where the user can increase or decrease the length of the vector, affecting the force applied to the movement. This force is always applied in the basket ball's center of mass.

As the user begins moving the vector, he will notice that a red line is coming out of the original ball, as it happens in Figure 4.6. This line is the predicted **trajectory** the ball will have with the current vectorial velocity in the hologram. This trajectory will also take into account eventual hits on different surfaces, like the basketball board or the floor.

At the bottom of the hologram, there are two different sections: to the right, some text cues are given in order to help the user take the right decision: the ball's current scalar velocity, and the scalar velocity that will be given by the hologram at the vector's current position. By joining this

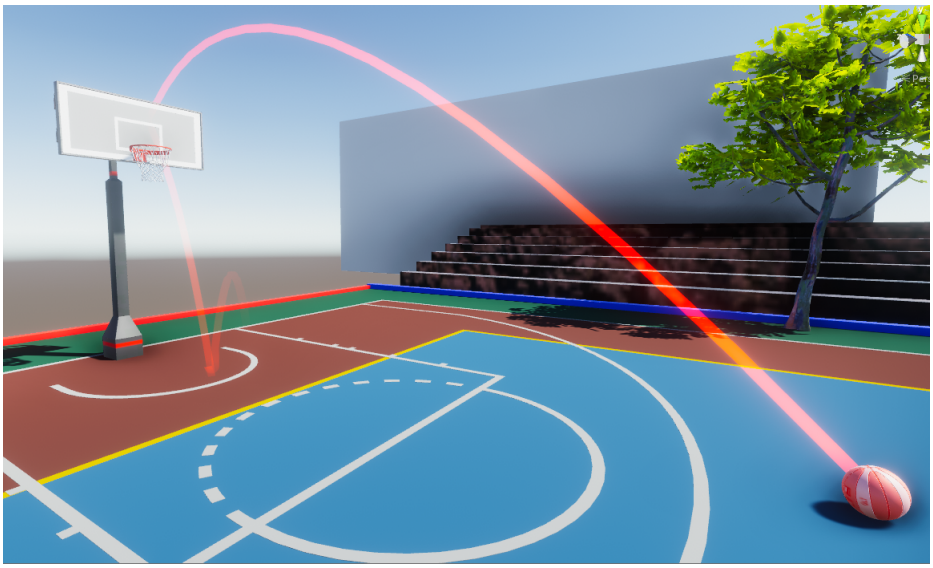


Figure 4.6: Representation of the trajectory curve, including bounces.



Figure 4.7: Detail of the axes and the velocity text cues.

information with the visualization of the trajectory, the user can make a more conscious decision on where he will throw the ball. To the bottom left of the hologram there is colored axis. This axis serves just as a visual indication of which direction the ball will go, and is colored to follow along with Unity's 3D coordinates system, where Red corresponds to the X coordinate, Green to the Y coordinate and Blue to the Z coordinate. Both these cues can be visualized in Figure 4.7.

For translating the desired movement of the hologram to the ball, the user simply has to press the trigger button on either controller, which will apply the velocity of the hologram's vector to the original ball.

When the ball is moving, not only the user has the textual feedback of the ball's current velocity, but a new, smaller vector also appears inside the hologram. This new vector is the ball's current vectorial velocity, and will follow the ball's movement until it stopped. This vector can be seen in Figure 4.8.

One of the main difficulties found on early designs of the prototype was that the users found it very hard to direct the ball in the direction wanted. The hologram uses the world's coordinates

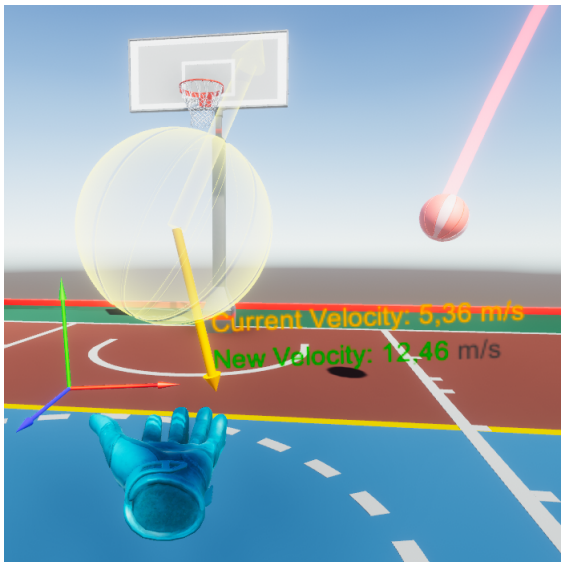


Figure 4.8: Current velocity vector (at yellow) and textual cue.



Figure 4.9: Red and blue guidelines around the edge of the basketball court.

instead of local ones, and as such, if the user wants to send the ball to the left of him, pointing the hologram's vector to the left may not be the right direction. Taking this into consideration, the coordinate axes were implemented to help guide the user. But it wasn't enough as the users still didn't know where the red or blue axis direction was. To tackle this problem, some coloured guidelines were implemented along the edge of the basketball court, representing the X and Z axes, and coloured red and blue, respectively, to follow along with the pattern used in the hologram's axes. These guidelines can be seen in Figure 4.9.

Another difficulty was that, since the basketball hoop's net is not animated, sometimes it was hard to tell whether the user had scored a shot or not. As such, a simple particle system was put in place, along with a brief audio clip to signal if the shot taken was successful or not. An example of this can be seen in Figure 4.10.

After some time in the experience, the user may not keep track of every shot he made, nor



Figure 4.10: Simple effect for when a user scores a shot.

the circumstances of every shot. As such, a **scoreboard** was also placed in one of the basketball court's corners and it displays various statistical data about the user's performance. This data was divided into three categories:

- Stats - The Stats category shows, on the left side, the total points the user has scored, the ratio of scored shots to attempted shots, and the hit scored shots percentage. The right side shows the details of the last shot attempted, with the distance it was taken from, and the velocity that was imprinted in the shot. An example is shown in Figure 4.11a.
- Last 6 - This category shows more details about the last six shots taken. For each shot, it is displayed if it hit, missed or if it was close to hit, along with a coloured feedback, for easier and more instant recognition of the state. These colours range from red, to orange, to green for the states of missed, close and made, respectively. Next to the state it is displayed the distance from where the shot was taken, followed by the scalar velocity of that shot. An example can be seen in Figure 4.11b. This section of the scoreboard is intended to give a small summary of his last actions in the experience.
- Missions - This section was divided into two scoreboard screens, for better readability, as they would take too much space in one single screen. This section is displayed in Figure 4.11c. In both screens, all the missions the user has to make are displayed. Each mission is coloured red if it is not accomplished yet, or green if it has been accomplished. At the end of each mission there is also a marker to identify how many shots were made and taken, in the form of a ratio. For example, showing *(2/4)* means the user attempted four shots for that mission and hit two of them.

The scoreboard automatically switches between each section every five seconds, always in the following order: *Stats - Last 6 - Missions (Part 1) - Missions (Part 2)*. After reaching the last section, it returns to the first one.

The wind feature also required some visual feedback for the user to easily understand in which direction the wind force was headed. As such, a scenario where the leaves of a tree are blown with the wind was thought of being a realistic one, and one that would help with the user's immersion.



(a) Stats panel. (b) Last 6 panel. (c) Missions panel.

Figure 4.11: Different panels of the scoreboard.

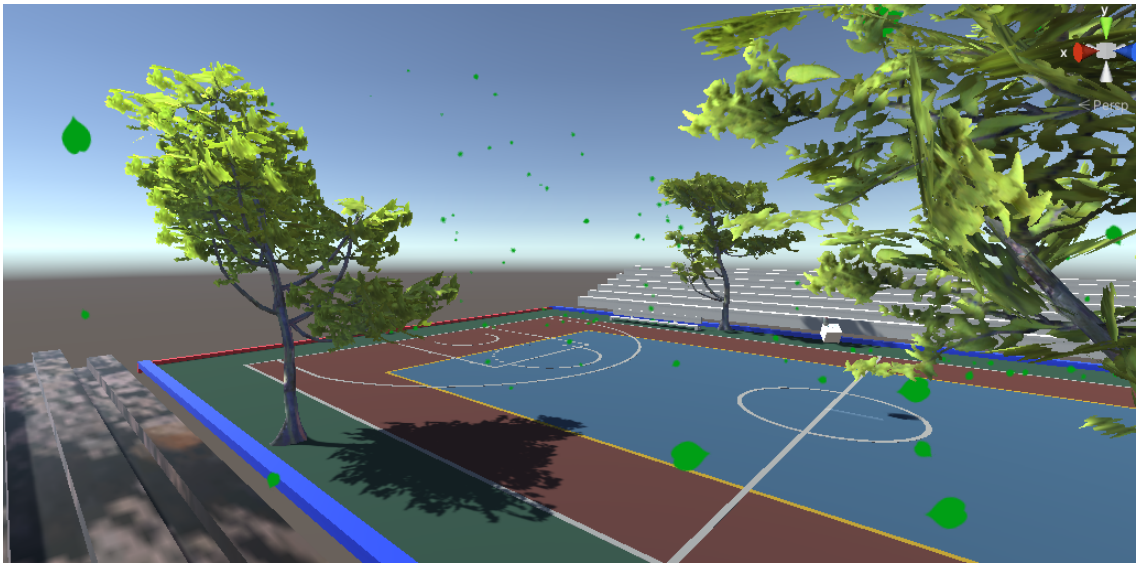


Figure 4.12: Leaves blown to indicate the wind direction.

The leaves are thrown in the direction specified, and will travel further if the wind intensity is greater, and can be seen in Figure. The trees will also shake more violently if the intensity is greater.

To have a more realistic touch to this mechanic, the leaves do not move all in same direction as it is experienced in a real life context. When the wind blows in one direction, each leaf behaves as a unique object and has slight variations towards other leaves in proximity. To simulate this situation, an area was defined, limited by a cone of action, inside which the wind acts on the leaves, as seen in Figure 4.13.

Another important feature about this dissertation is the ability to stop and resume time at will. This is done with the push of the button "Y" on any controller. For this dissertation, only the basketball is configured to respond to the time control feature. Using this feature will stop the ball in the exact position when it is activated. After this, the user will be able to see two distinct

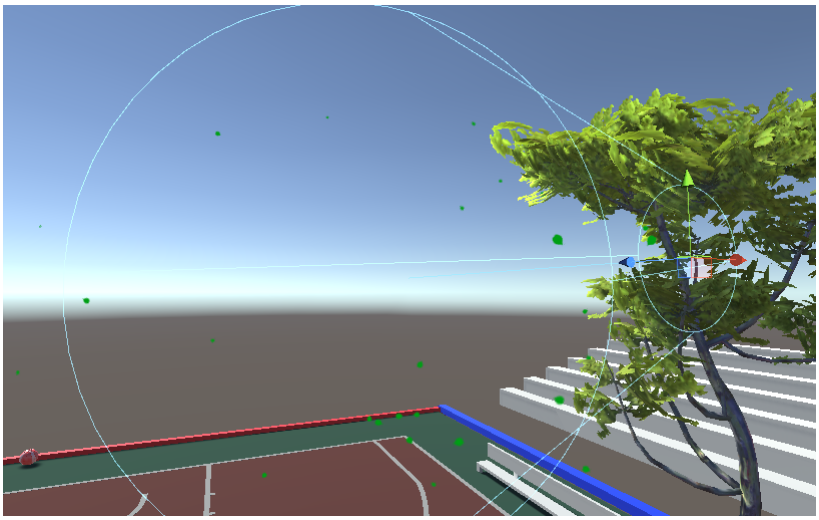


Figure 4.13: The direction of the leaves will be randomly selected, but limited to a direction that is inside the cone.

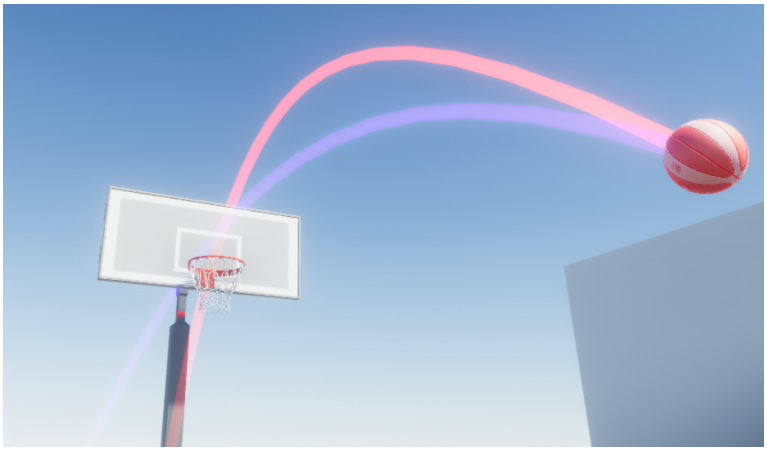
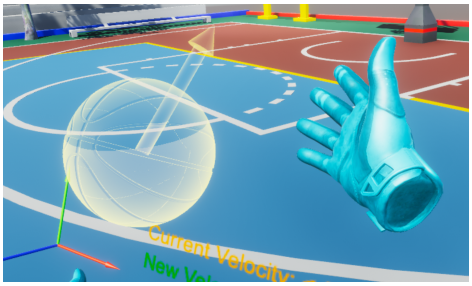


Figure 4.14: Using the time control will spawn the blue trajectory, representing the course the ball will have if the motion is continued.

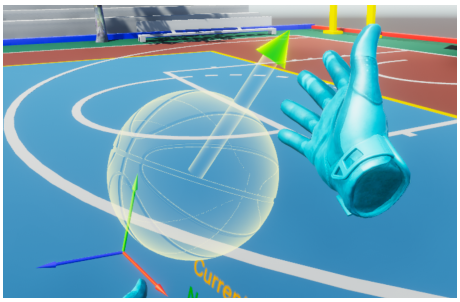
trajectories: a blue one, which is the movement the ball would have if it wouldn't be stopped; and a red one which is the new one if applied with the hologram, as it happens with a normal shot. Both trajectories can be seen in Figure 4.14. With time stopped, the user will have two options: he can either let the ball retake its original movement, effectively resuming time, which is done by pressing the same button in which he activated it; or choose another movement, by selecting a new velocity vector in the hologram, and using the trigger button to send the ball in the new direction.

4.4 VR Interaction/Locomotion Techniques

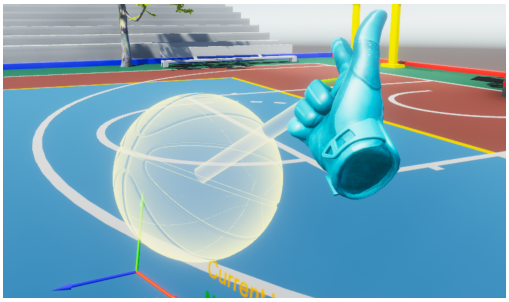
Interaction and locomotion are key figures when it comes to VR applications. This dissertation is no different, and an increased effort was given when designing these aspects. Reviewing many of the techniques researched in section 2.1.2 and section 2.1.3 inspired the creation of the hologram



(a) The hologram in its "normal" state, with the hand close to it.



(b) The hologram's vector tip turns green to signal that it can be grabbed from that position.



(c) The hologram vector is grabbed.

Figure 4.15: Steps of grabbing the hologram.

technique mentioned before, which could be categorized as a hybrid between a grabbing technique and a at-a-distance technique.

Interacting remotely with the basket ball is a critical component on this work, as it must allow the user to move it in any direction, and with a specified velocity. Taking this into consideration, a representation of the ball was created and attached to the user's left hand, in a form of a **hologram**. This representation simulates only the rotational movement of the ball, so that the user gets a visual stimulus that the movement of the original ball is synchronized with the hologram one. An arrow is appended to the hologram, fixed to the center point of the hologram and represents the vectorial velocity of the desired movement of the ball, in a 3D world space. The user can grab the tip of this vector to control the movement of the ball. In order to facilitate this grabbing mechanism, when the user's right hand is in grabbing position, the tip of the arrow turns green, to symbolize that an action can be taken. This process is illustrated in Figure 4.15. With the vector attached to the hand, the user can move it around the hologram to define the ball's next movement. The vector movement must be decoupled into two phases: rotation and scaling. Rotating the vector around the hologram controls the direction of the movement the ball, relative to the world space. An axis is displayed below the hologram to help the user identify the direction of the desired movement, as well as some guidelines along the edge of the basketball court.

The hologram's grabbing mechanism suffered a number of iterations during development until its final state. Initially, the user's right hand had a snapping point in its center, to which the hologram's vector would snap into when it was grabbed, matching the tip of the vector with the hand's snapping point. This mechanism would work well to move the vector around the hologram

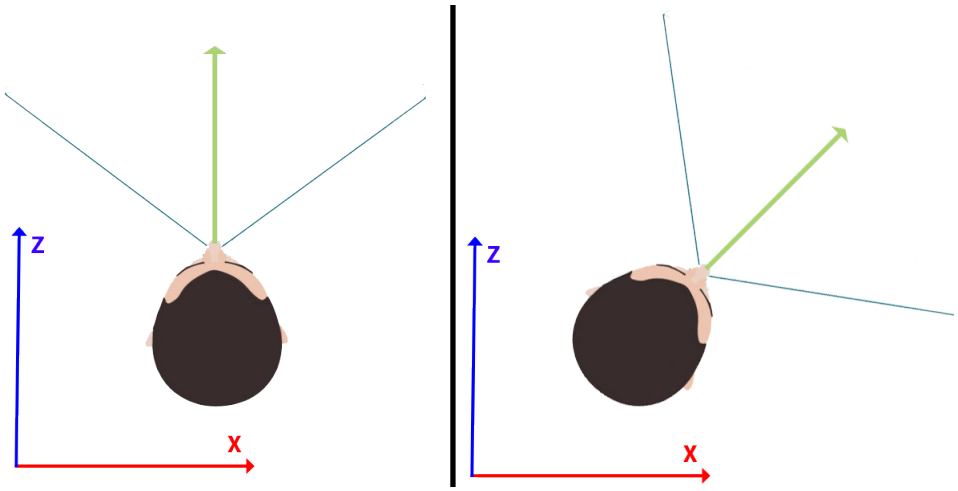


Figure 4.16: Forward joystick movement. The user will move in the direction of the green arrow.[15]

but it had one major fault. It was extremely hard to position the vector in the same place twice. Since most of the times a user would grab it was to make precision adjustments, snapping the vector to the hand by making it move to encounter the hand would throw off this precision. It was eventually settled by doing this mechanism reversed. Instead of the vector snapping to the hand, the snapping point became a **pivot point**. It was initially placed in the center of the user's right hand, but after the user grabbed the vector, this pivot would move to the exact position of the tip of the vector, instead of the vector moving to the hand. By doing this, the vector would not move when it was grabbed and allowed for a greater precision on the movements. Since the pivot point was still attached to the hand, moving the hand after grabbing would move the hologram's vector as well. This method also solved a problem that the user can grab the vector from any direction or orientation of the hand, since the pivot would adapt to where the end of the vector is.

The final addition to the grabbing mechanism was a stabilizing factor. It was noticed in testing phases that users would slightly shake their hands when making precision adjustments, and this shakiness would difficult precision control, such as when they were targeting the basketball hoop. To counter this effect, a stabilizing factor was added, meaning that at each movement of the hand, it is calculated the distance from the previous hand position, and if this distance is shorter than the stabilizing factor, the movement is ignored. This helped to minimize the shakiness of the hands, and although it did not completely remove it, it helped to achieve a better control over the hologram.

Regarding locomotion, a joystick-based technique was implemented as the main locomotion method for this prototype. Taking advantage of the tested HMDs controllers (Oculus Rift and Oculus Rift S), the joystick was configured to move the user in the direction it is tilted relative to the user's field of view. If the user would move or rotate his head, a new direction is calculated based on the new position and orientation, making this method independent from the world axis. For example, a continuous forward motion in the joystick will move the user in different directions, according to head orientation, as can be seen in Figure 4.16.

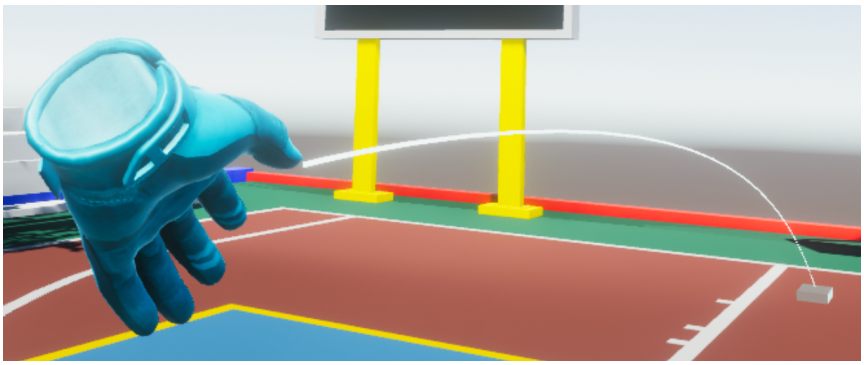


Figure 4.17: Using the teleportation technique.

Even though this technique works as intended, some people report having increased sensations of motion sickness using joystick-based locomotion. Therefore, it was important to add at least a second option, if someone was feeling uncomfortable using the first one. The teleportation-based technique tackles this problem well, and was chosen to be added as the second locomotion method. Still, since no menus were planned to be implemented, some method of incorporating both techniques was needed. As so, the joystick was once again used for this method. By pressing the joystick and then moving it, it activates a parabolic trajectory, and the user will see where he will teleport. The user just needs to point where he wants and let go of the joystick, and he will go to the position pointed. This method works without conflicting with the first one, as it needs the joystick to be pressed, instead of just pushing the joystick. The teleportation technique was imported from the project developed in Pereira's [Per19] work and adapted to be compatible with the joystick-based one. This technique can be seen in Figure 4.17.

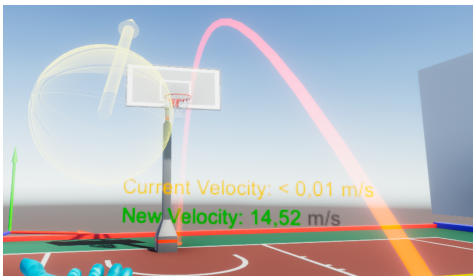
Another useful locomotion tool is the ability to rotate a certain number of degrees (45 or 90), as the user may not have the physical space in its room to do so. While this snap rotation feature was considered for this project, it ultimately was not implemented as it was deemed not essential.

Although no interacting menus were implemented in this prototype, the scoreboard consists of a world fixed system, that shows information in a strategic position, and stays in that same fixed position throughout the whole experience. The hologram's textual cues are an object fixed system, as they are attached to the hologram (which in turn, is attached to the hand), and follow along with the hologram's movement, as seen in section 2.1.2.3.

4.5 Physics simulation

To allow for the representation and interaction with a system that simulates some physics laws is one of the main goals in this project. This simulation was constructed considering the VR interaction and locomotion techniques described in section 4.4 and attempts to combine these techniques to seamlessly integrate with these physics concepts. In particular, the hologram manipulation

[15] Image adapted from https://vr-lens-lab.com/field-of-view-for-virtual-reality-headsets/, last access June 2020



(a) The hologram is positioned in front of the basketball board.



(b) The hologram is positioned laterally of the board, with no change in the trajectory.

Figure 4.18: Different perspectives of the same hologram vector, giving the same result.

needs to follow concrete laws so that the ball's movement is believable and does not behave unexpectedly. Additionally, the precision of the trajectory also needs to be carefully calculated, as a minor imprecision may throw the ball off course, and cause the user to miss the target.

The time control mechanism, although obviously not realistic, needs to keep track of the velocities of the stopped objects, so they can be resumed, if that's the case. The wind force was also simulated, and although it was simplified for the scope of this project, still obeys a level of realism making it become believable.

4.5.1 Hologram Manipulation

The hologram's vector is the one that dictates the direction and velocity of the movement of the ball. When the hologram's velocity is applied, the ball's vectorial velocity is immediately substituted with the one of the hologram's vector. A prior version of the hologram added the vector's velocity to the ball instead of substituting it, but it was considered to have a lesser degree of control, since the velocity the user would see in the hologram would not correspond to the ball's effective velocity, and to represent that, another vector should be included, representing the sum of the velocities.

Initially the vector has a scale of one unit, corresponding to a velocity value of 10 m/s. Scaling the vector follows this linear scale of one unit to 10 m/s, up to a maximum of four units, or 40 m/s. The vector can't also be scaled to negative values, as trying to scale it down towards negative values will rotate it to the opposite direction.

This vector values are relative to the world coordinates, which means that if the hologram's vector is turned exactly forward, with a magnitude of one, its value will always be (0,0,1), independently of the position or orientation of the user. In Figure 4.18 the hologram vector is the same in both images, and even though it can be seen from different perspectives, it translates to the exact same velocity to be applied to the ball, as there was no change to the direction of the vector in relation to the world's coordinates.

4.5.2 Ball Trajectory

The basketball's trajectory is an estimation of the movement the ball will have, according to the laws of projectile motion. For this trajectory to be drawn, an arc is seen exiting the center of the original ball, and will behave accordingly to the current velocity set on the hologram's vector. For this to happen, two parameters must be calculated beforehand: position and velocity. The estimated position of the ball is calculated according to equation 4.1:

$$\vec{p} = \vec{p_{ini}} + \vec{v}t + \frac{1}{2}\vec{g}t^2 \tag{4.1}$$

In this equation, p is the new position, p_{ini} is the previous position of the ball, v is the current velocity of the ball, t is the current time, and g is the vector of acceleration, which in this case corresponds to the gravity value, as there is no other force involved. This gravity vector takes the default value of *(0,-9.8,0)* as an approximation to the world's gravity force, but can be changed in the external settings, as explained in section 4.7.

But in order to calculate the position, there is the need to know the vectorial velocity of the ball in each instant, which is calculated according to the following equation:

$$\vec{v} = \vec{v_{ini}} + \vec{g}t \tag{4.2}$$

In equation 4.2, $\vec{v_{ini}}$ stands for the velocity of the ball in the previous instant.

During initial testing phases of the trajectory, using these calculations did draw a similar projection to the actual real trajectory of the ball, but it was noticed a slight difference, that would increase as the distance and velocity of the shot also increased. Although this difference was small in distance, it would greatly affect the user's performance, as just few centimeters away from expected can mean that the shot will be missed. This was happening due to the **drag** effect, where the air resistance would hinder the progression of the ball and slightly reduce its velocity. The drag force acts in direction opposite to the object's motion and is proportional to the force. However, and in accordance to Unity's simulation of the drag force, the ball's velocity was recalculated according to equation 4.3.

$$\vec{v_{fin}} = \vec{v}(1 - dt) \tag{4.3}$$

In here, $\vec{v_{fin}}$ is the final velocity value, $\vec{v}$ is the velocity value calculated in equation 4.2, and d is the drag force generated on the ball, which was left as the Unity's default value of *1*. Additionally, the expression $\vec{v}(1 - dt)$ was limited to values between *0* and *1*, as the drag force is only able to reduce the projectile's velocity, and not increase it or even change its direction.

With the adjustment of the equation 4.3, the observed trajectory seemed to be on par with the ball's actual trajectory, with no apparent errors even at greater distance or velocities. The trajectory was also configured to reflect eventual bounces on various surfaces like the floor or the basketball board, and how it will behave accordingly.

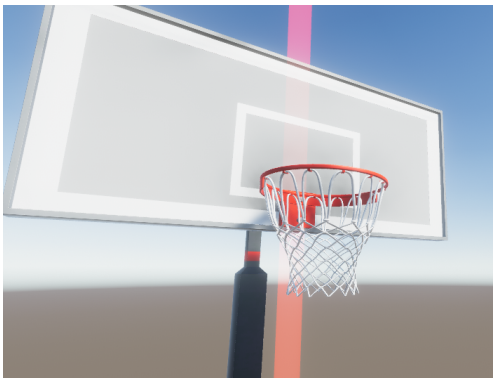


Figure 4.19: Trajectory makes it look like it will be a successful shot, but the ball will actually hit the side of the rim and bounce out.

However, it should taken into consideration that these calculations are an approximation of the trajectory. An example where the trajectory may lure the users into error is the one depicted in Figure 4.19, where the trajectory shows that the ball will enter the hoop. This will in fact, not happen, as the diameter of the ball is considerably greater than the one used in the trajectory arc, and as such, the ball will actually hit the side of the rim, bouncing off the hoop and missing the target.

4.5.3 Time Control

As seen in previous sections, the time control mechanic is one where the user can stop and resume time at will, with the possibility of changing the ball's trajectory mid-course. In order for this to happen, when the user sends the command to stop time, the objects that can be affected by it have their velocities immediately turned to zero, while also disabling their interaction with gravity forces. This enables specific objects to stand still in the air, instead of pausing the whole simulation, as that could have unforeseen consequences. At this moment, the objects' previous velocity is stored internally, so it can be accessed later.

The user is then provided with two options: he can either let the object continue its previous trajectory, where the velocity that was previously stored is returned to the object, or he can set a new velocity with the hologram, and the object will follow its new course. In this last case, the stored velocities are ignored and discarded, as they will not be needed anymore.

4.5.4 Wind

For the wind simulation, a simplification of a real wind effect force was used. Instead of a constant force in a direction, and in order to simulate gusts of wind with variable force, a maximum wind intensity value is indicated in the external settings file. The actual wind intensity is calculated at every instant with a random value between zero and the maximum intensity value, and then added to the affected object's velocity according to the following equation:

$$\vec{v} = \vec{v_{ini}} + (\vec{wd})i \tag{4.4}$$

In equation 4.4, $\vec{v}$ is the object's velocity, $\vec{v_{ini}}$ is the object's initial velocity without any wind force affecting it, $\vec{wd}$ is the wind direction (this vector will always have a magnitude of one, to not interfere with the intensity), and i is the wind intensity. This force is added to the ball's velocity vector, thus simulating that a force is effectively pushing the ball in a certain direction.

4.5.5 Shot detection

Detecting if a shot was taken and verifying its outcome was needed to assess the user's performance. In order to detect if a shot was in fact taken, and not just an adjustment of position, a spherical area was created around the basketball's hoop, and only if the ball entered this area during its flight it was considered that the user was attempting to shoot the ball. This area can be seen in Figure 4.20a. A similar approach was followed for classifying the shots, as another spherical area was created for this purpose, although significantly smaller than the previous one, as seen in Figure 4.20b. Three different classifications were created: missed, close and made. If the shot only touched the the bigger area during its course, it was classified as a miss. If the shot eventually entered the smaller spherical area, then it was reclassified as a close shot.

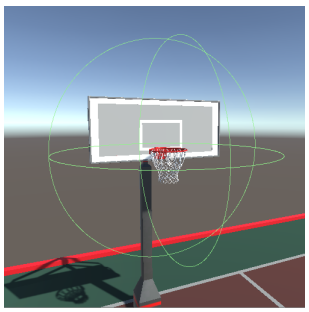
Finally, two cuboid-shaped areas were created on the top and bottom part of the basketball hoop as can be seen in Figure 4.20c. A shot is reclassified as successful only when it passes the top area and then the bottom area, in this order only.

4.6 Collaboration and Networking

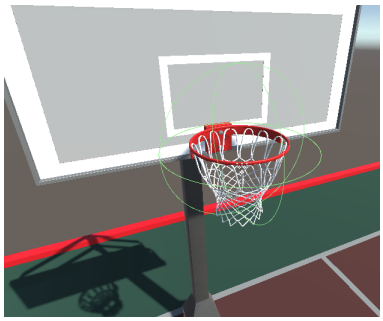
In addition to the representation of the physics laws involved, the collaboration and interaction between different users are one of the key concepts in this project. This feature allows that two users interact with the system where each of them has different functions in the scenario in order to make that only by working together and complementing each other actions allows them to achieve the desired outcome. This collaboration is limited to two users at a time and the way this process was structured and its communication were implemented are detailed below.

4.6.1 Asymmetric Collaboration

As stated in section 3.2, one of the desired requirements for the project was the inclusion of asymmetrical mechanics for the collaboration aspect. By providing the users with different ways to interact, it was intended to increase a greater sense of purpose and collaboration between the users. In order to achieve this, it was decided to create two different roles for each user. The **manipulator** will be the one that has control over the hologram, and as such, will indirectly control the basketball as well. But, contrary to the single-player experience the manipulator will only see 10% of the basketball's trajectory (see Figure 4.21a). This value was achieved after several iterations on the design. The first implementation of the roles had the manipulator see no



(a) The area the ball must go through to be considered an attempt.



(b) The area the ball must go through to be classified as a close shot.



(c) The two cuboids the ball must pass to be considered a successful shot.

Figure 4.20: Shot classification areas.

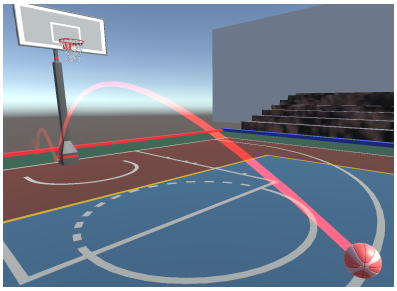
trajectory at all. But in early testing runs, it was seen that the users that would take the role of a manipulator would struggle too much to align a correct trajectory. By giving them just a slight visualization of the intended path, the manipulator can at least align the general path he wants, and calibrate it with the feedback from the other user.

This other user will take the role of the **observer** and as such, he will have no control whatsoever on the hologram. Instead, his role is to inform the manipulator if the current trajectory is in fact aligned with the target or not, and how he can adjust it to be successful. Contrary to the manipulator, the observer sees the full trajectory of the basketball 4.21b, according to the direction the manipulator is currently setting in his hologram. The same happens when the manipulator uses the time control mechanic, as the manipulator will only see 10% of both trajectories (see Figure 4.21c), the observer will see both full trajectories (see Figure 4.21d).

It was also implemented a method that the users would switch roles after a number of successful shots, taking turns with each role, so they could experience both sides of the experiment. However, since the objective of this work is to assert the effectiveness of the hologram manipulation and collaboration between the users, it was decided to turn this feature off to the final experiment, as it would add an extra layer of complexity, since all the users would need to learn to take two different roles. This would also imply the need to evaluate the effectiveness and performance of the user that is communicating, which is something that is out of the scope of the project.



(a) The trajectory seen by the manipulator in a multiplayer setting.



(b) The trajectory seen by the observer in a multiplayer setting.



(c) The trajectory seen by the manipulator after using the time control feature.



(d) The trajectory seen by the observer after using the time control feature.

Figure 4.21: Perspectives of the same shot for the manipulator and the observer.

4.6.2 Sending and Receiving Signals

Since early iterations on the collaboration aspect for this dissertation, it was decided that there should be some form of non-verbal communication between the users. Ideally, it would totally substitute the need for any kind of verbal interaction between them, either by visual or auditory feedback. However, this was eventually substituted with a way to complement verbal communication, giving the users an advantage to use it instead of just talking to each other. This mechanism consists on the observer sending **signals** to the manipulator. In his left hand, the observer has three different buttons in place of the hologram, each corresponding to a different signal to send to the manipulator as can be seen in Figure 4.22. The observer can use these buttons to tell the manipulator that the current trajectory is too far from the target (red button), that it's close to the target (yellow button) or that it's exactly on target (green button).

On the other hand, when the manipulator receives these signals, his hologram will instantly start flashing in the color of the button the observer pressed. The biggest advantage for the manipulator, is that not only he knows that the current trajectory may be off/on target, but a **snapshot** of the hologram's current velocity vector is taken, and left in the hologram, coloured by the respective signal. These snapshots allow for a significant greater control over the trajectory, as the manipulator can guide himself not only on verbal communication with the observer, but also by seeing in his own hologram which trajectories are off target and align a new one according the snapshots he sees. The number of snapshots that can be taken was left for the experiment phase



Figure 4.22: Three coloured buttons, for the observer to send signals to the manipulator.

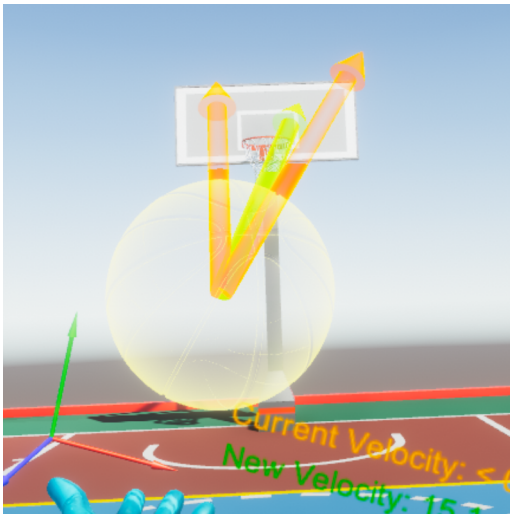


Figure 4.23: Three snapshots, sent from the observer. Two red and one green.

as three, as having too many snapshots in a small area would turn to be confusing to the user manipulating the hologram. An example of these snapshots can be found in Figure 4.23.

4.7 External Settings

One of the requirements listed in section 3.2 was the ability to read certain world parameters directly from an external file. This allows for a greater control over the conditions present in the experiment without the need to change the internal code or the object structure of the project.

For this settings file, the *.json* file format was chosen for its simplicity and being easy to edit and understand.

This feature was implemented with the maximum flexibility as possible. Some of the parameters thought possible were implemented, but there is room left for many more that would take minimal programming effort. For the scenario chosen in section 4.1, it was decided to implement two of these world parameters via an external file: the world gravity and a wind force. The following is an example of a full settings file, and its details will be further explained in this section.

```
{
  "ball": {"gravity": "4.9"},
  "wind": {
    "intensity": "0.5",
    "direction": {
      "x": "0.5",
      "y": "0",
      "z": "1"
    }
  }
}
```

4.7.1 Gravity

Unity uses by default a simulated gravity with an approximate gravity downwards acceleration force of 9.8 m/s^2, which leads to a very feasible physic simulation. Nevertheless, it was considered what could happen if this value of gravity would change, and how the users would adapt to these circumstances. However, and even though Unity easily allows to change this force value, changing the entire scene gravity could lead to unintended results. As an example, lowering or increasing the world gravity would lead to the users moving slower or faster than normal, which could lead into greatly increasing feelings of VR motion sickness. Since this would be a hard problem to foresee, as there could be many objects in the scene that could negatively affect the performance of the users under such circumstances, it was decided that this shift of gravity would only be allowed to specific objects. As such, for an object to be affected by a different value of gravity force, the object name should be provided as the *name* in the *name/value* pair in the file. As for its *value*, another *name/value* pair is added, with the *name* being the string *"gravity"*, and the *value* being the force value wanted for the gravity value. This method of organizing data allows for other parameters to be changed on only that specific object, such as its weight, although no more were implemented in this prototype. An example of using this feature to change the gravity is the following:

```
{
  "ball": {"gravity": "7.1"},
}
```

This example will change the gravity value for only the *ball* object to 7.1 m/s^2. In this prototype, the basket ball is the only object that can be affected by gravity changes, down to a minimum of 1 m/s^2. If no value of gravity is presented, or if there is an error in the file, the application will use the default gravity acceleration force value of 9.8 m/s^2.

4.7.2 Wind

The wind feature that was explained in section 4.5.4 is also defined in the external settings file. Under the *"wind"* keyword, an object with two parameters is included: intensity and direction. The intensity is the absolute value of the force the wind will imprint in the affected objects, and needs to be defined by a value greater or equal than 0. The direction is the direction of the force that affects the objects in a 3D world coordinates. These values can be either negative or positive to reflect all possible axes. Although the absolute value of each component of the direction can be greater than 1, the value will be normalized afterwards in order to get a total length value of 1. This vector will then be multiplied by the intensity value to get the final wind velocity value. In the example below, the wind velocity correspond to a total of (0.3, 0, 0.9).

```
{
  "wind": {
    "intensity": "1",
    "direction": {
      "x": "1",
      "y": "0",
      "z": "3"
    }
  }
}
```

4.8 Retrieving analytics data

The analytics manager plays the final and vital role of the prototype. It allows to extract all relevant usage data from the last experiment, analyze it and export it to an external file to be analyzed later. The data that is actually extracted from the usage consists in the shots that are taken from the beginning of the experiment phase until the end, but not during the freeplay phase (these phases will be further detailed in section 5.1). In order to understand how the data is structured, it is important to understand the following concepts: a **Shot** is considered as an attempt a user makes of sending the ball to the basketball hoop, and it can either miss, be close to hitting, or hit the target; a **Grab** is considered as the action the user does when he actually grabs the hologram's vector until he lets go of it (not until he shoots the ball). A *Shot* can have multiple *Grabs*, but it will always have at least one. A **Signal**, as explained in section 4.6.2 is the communication the observer makes with the manipulator, and can be either a green signal, a yellow one or a red one. A *Shot* will only have multiple *Signals* in a multiplayer context, as there is no communication in the single-player experience. A **Time Control Usage** is the action of stopping the time, and as a consequence, stopping the ball in mid-air, and then resuming time by letting the ball take its original course, or imprint a new velocity and direction on the ball, as detailed in section 4.3. A

Shot will have either zero or one *Time Control Usage*, as the user was allowed to use this ability only once per shot taken.

For each *Shot* taken in the experience, the following data is collected:

- Distance - The distance from the basketball original position to the basketball hoop
- Velocity - The velocity that was applied to the ball, in its vectorial form
- Time - How much time has elapsed since the last shot
- Shot status - The result of the shot: if it was missed, if it was close to hit, or if it actually hit
- Grabs - All *Grabs* that were made in that shot.
- Signals - All *Signals* that were sent from the observer related to this shot, if any.
- Time Control usages - All the usages of the time control mechanic, if any.

The *Grabs* were subdivided into various components that could be analyzed.

- Initial Velocity - The velocity of the hologram's vector in the instant it is grabbed, in its vectorial form.
- Final Velocity - The velocity of the hologram's vector in the instant it is dropped, in its vectorial form.
- Velocity difference - The difference between the final and initial velocity.
- Time - The time spent since the hologram's vector was grabbed until it was dropped.

As for the *Signals*, for each one sent in the shot, the following data is retrieved:

- Velocity - The velocity the manipulator had in the hologram in the moment the signal was sent, in its vectorial form.
- Signal type - The kind of signal that was sent: red, yellow or green, as explained in section 4.6.2.

Finally, for the *Time Control Usages*, they were divided into the following components:

- Distance - The distance of the ball to the basketball hoop in the moment the time was stopped.
- Initial Velocity - The velocity of the ball in the instant the time was stopped, in its vectorial form
- Final Velocity - The velocity applied to the ball when time is restarted, in its vectorial form. This value will be equal to the Initial Velocity if the user did not correct the ball trajectory, and let the ball take its original course.

All this data is exported to a .json file when the application is closed, so it can be evaluated afterwards. An example of the data that is captured can be found in Appendix A.

4.9 Summary

This chapter began with an overview of multiple scenarios that could be implemented in this dissertation, along with the case study selected. The various tools and technologies used in the implementation were also detailed, along with the interfaces created for the prototype. The used VR interaction and locomotion techniques were detailed next, followed by the multiple physics simulation aspects concerning this work. Another important aspect was discussed in the networking and collaboration section, explaining the roles of the manipulator and the observer. The chapter finalizes with an explanation of how the external settings are defined and imported, and how the data is extracted from a user's test session.

This data will be essential to the evaluation of the prototype, that will be discussed in the next chapter.

Chapter 5

Evaluation

In order to evaluate the system, user studies were done and data was collected in two distinct ways: through usage data, retrieved during the experiment, and through user questionnaires, handled after the experiment. These will allow to parametrize different aspects from the user's experience, specially concerning the main aspects of remote object control and collaboration.

This chapter starts the evaluation protocol defined in 5.1, leading to how each test session was structured and conducted in section 5.2. The different types of data that were collected from each experience is detailed in section 5.3. These experiments result in the user studies in section 5.4, where an analysis of all data collected is performed, along with the insights it provides. Finally, a discussion about this analysis is presented in section 5.5, concluding with a summary of the chapter in section 5.6.

5.1 Evaluation Protocol

The hardware requirements for the experience consists of a laptop to run the experience, and at least one HMD. The laptop used ran the experience on an Intel Core i7-6700HQ CPU and a GeForce GTX 1060 GPU. The used HMD for the experience was the Oculus Rift S along with its motion controllers.

The experiment is divided into two main sections: a single-player experience, and a multi-player one. The single-player experience was focused on the hologram manipulation and time-control usage. For evaluating these parameters, the following tasks were defined:

- **Distance-based shots (SP-DBS)** - The user must shoot the ball from pre-defined positions, as follows:
  - 4 shots from less than 3 meters of distance from the basketball hoop.
  - 4 shots between 3 and 6 meters of distance from the basketball hoop.
  - 4 shots from more than 6 meters of distance from the basketball hoop

- **Velocity-based shots (SP-VBS)** - The user must shoot the from pre-defined positions, but with an additional restraint on velocity:
  - 3 shots with a velocity of less than 15 m/s
  - Repeat the 3 shots but with a velocity of more than 15 m/s
- **Time Control Shots (SP-TCS)** - The user must shoot the ball from 5 pre-defined positions, but must use the time-control mechanic once.

Each user made 12 SP-DBS, 6 SP-VBS and 5 SP-TCS, totaling 23 shots in single-player mode. The SP-DBS are divided into three sections of increasing difficulty, and allow to evaluate if the user has good control over the hologram. The SP-VBS are taken from a distance where 15 m/s is an easy mark to make that shot. Having the user to shoot at a value smaller than that speed will obligate the user to make a more straight shot, with less curve, eventually using the backboard. Having him take the same shot at more than 15 m/s, will make the user point upwards, and make a more curved shot. Finally, the SP-TCS will help to evaluate if the user can effectively use the time control mechanic to improve his performance. The positions of the shots taken in this task are the same as the last section of the SP-DBS task (but at a different order).

Next, the user will go on to the multiplayer part of the experiment. The volunteer would take the role of the manipulator, with myself being the observer. This option was taken as the priority of this study is to assess how the collaboration aspects influence the performance of the manipulator. To have a volunteer also as an observer would need an additional evaluation on the performance of the observer as well, as his performance would surely impact the manipulator performance. By establishing myself as the observer, although it doesn't mean that the manipulator will have the best possible results regarding collaboration, it does ensure that all the experiences had the same level of observer's performance. This linearity in the observer's performance allows for a fairer comparison of the manipulator's results. Due to only having one HMD available for the experience, the desktop mode of the application was used for the observer only, where he could walk around the court using the W, A, S and D keys, use the mouse to rotate his view, and use the 1, 2 and 3 keys to send green, yellow and red signals, as explained in section 4.6.2. The required multiplayer tasks were the following:

- **Distance-based shots (MP-DBS)** - The user must shoot the ball from pre-defined positions, as follows:
  - 4 shots between 3 and 6 meters of distance from the basketball hoop.
  - 4 shots from more than 6 meters of distance from the basketball hoop
- **Time Control Shots (MP-TCS)** - The user must shoot the ball from 5 pre-defined positions, but must use the time-control ability once.

With the help of the observer, the volunteer will make 8 MP-DBS and 5 MP-TCS, totaling 13 shots in a multiplayer setting. The MP-DBS are divided into two sections, similar to the last

two of the SP-DBS task, but with the difference that he will only see 10% of the trajectory, and will have the help of the observer, as was explained in section 4.6. The aim of this task is to assess the effectiveness and engagement of the collaboration aspects, maintaining the same level of interaction with the hologram. Because of limited visualization of the trajectory, this task is considered to be of a higher degree of difficulty when compared to the SP-DBS task. The MP-TCS task will continue to assess the same parameters as the MP-DBS task, but with the increased load of having to use the time control mechanism. This will potentially create the need for the observer to help the manipulator twice, with the first shot not needing to be so precise, as it can be adjusted mid-air. Depending on where the manipulator stops the ball, the help of the observer in the second shot may not be needed.

Combining both single-player and multiplayer sessions, each user will make a total of 36 shots, as can be seen in the tasks overview in Table 5.1.

Table 5.1: Tasks overview and number of shots taken per task

Setting	**Distance (DBS)**			**Velocity (VBS)**		**Time Control (TCS)**	**Total**
	< 3m	3m - 6m	> 6m	< 15m/s	> 15m/s		
Single-player	4	4	4	3	3	5	23
Multiplayer	-	4	4	-	-	5	13

5.2 Test Session Structure

The tests began by giving a short explanation on the HMD used, the Oculus Rift S and its motion controllers. The users were then instructed on how the test will be conducted, how each task will be handed to him, and that after the single-player tasks, I would join him inside the experience to help him through the rest of the experience as an observer. The user was informed that the tasks were also available for him to track in the scoreboard.

Each user was also informed that they could ask to end the experience at any time, for any reason, specially if they started to feel any kind of discomfort or nausea using the VR system.

In each session, before the experience began, the volunteers would also be asked to thoroughly read an Informed Consent document, which required their signature to proceed with the experience. This consent can be found in Appendix B.

It was also explained that the application will start with a free-play session, that would work as a tutorial for the needed controls for every mechanic, and that no data would be collected in this phase. Only when they would feel comfortable and confident with the controls would the experience begin, which will then be collecting data about the usage, but in no way they would be asked for input about personal or private information.

If no further questions were asked, the user would then place the HMD in their head with the application already started. A free-play session began, and the users were informed on the various controls: first on how to move around the world using each type of locomotion, and how to interact

with the hologram and time control. A brief multiplayer test session with myself would then take place, in order for the user to see the differences of multiplayer versus single-player, regarding the visualization of the trajectory and the vector snapshots the observer would send.

After the volunteer informed they feel comfortable and would like to begin the experience, it would start by pressing the Spacebar key on the laptop's keyboard.

The single-player tasks would be conducted, and after that, the volunteer was informed that I would now enter the experience in their virtual world, and that they should look for my avatar somewhere in the basketball court. After recognizing my avatar, the multiplayer tasks would take place.

Finalizing the multiplayer tasks, the volunteer would be asked to remove his HMD, and to start filling the set of questionnaires prepared for the session. These consisted on a basic demographic questionnaire, a Simulator Sickness Questionnaire [KLBL93], a general System Usability Scale questionnaire [WD17] and finally, some Likert Scale questions regarding the experience and interaction with the system.

Table 5.2 gives an summarized look over the test structure.

Table 5.2: Structure of each test session

1. Introduction to VR and to the experience		
2. Participants signed the Informed Consent		
3. Participants played the experience	Single-player	DBS VBS TCS
	Multiplayer	DBS TCS
4. Participants filled the System Usability Questionnaire		
5. Participants filled the Simulator Sickness Questionnaire		
6. Participants filled the questionnaire about the Single-player and Multiplayer experience		
7. Participants filled the demographics questionnaire		
8. Participants were asked for general feedback		

5.3 Available Data

In every test session, there will be two types of retrieved data: the user questionnaires, where the user will state his opinion about the experience, collaboration aspects and interaction points. Additionally, the system will also automatically collect usage data from the hologram manipulation, the shots taken, and the collaboration between the users.

The usage of both these methods allows the evaluation of subjective opinion about the system and also a more analytical one while considering the users have consented the data collection but are not aware in which moments it is being taken.

5.3.1 User questionnaires

The experience of the users was collected in a questionnaire with a set of questions for the single-player section of the experience and other for the multiplayer section. Each question answer options was defined according to a Likert scale ranging from 1 to 5 points, in which 1 corresponds to "Strongly disagree" and 5 to "Strongly Agree". Additionally the user had also the opportunity to had a comment about his or her experience. The feedback from the users' experience is relevant to analyze the platform created and to consider the possible future improvements.

5.3.2 Experiment Usage Data

Throughout the whole experience, the system captured a number of parameters relevant to assess how effectively the participants used the various mechanics of the prototype, focusing on the hologram manipulation and the participants collaboration. The main aspects to be studied from these logs, whose creation can be found explained in section 4.8, are the following:

- **Distance** - The distance of the shot taken. Distance is calculated as a straight line between the basketball's current position, and the basketball hoop, as can be seen in Figure 5.1.
- **Velocity** - The initial velocity vector imprinted on the ball. This value is important to establish an ideal velocity the users should throw the ball. In a basketball game context, from one fixed position, there are many ways to score the ball, resulting of numerous combinations of the angle between the floor and the velocity vector, with the force the ball is thrown.
- **Grabs** - By checking the number of Grabs made during a shot, we can see if the level of difficulty that shot had on the user, as more Grabs will indicate that the user needed to correct the ball's trajectory more times.
- **Time spent** - The time spent between each shot will help to correlate the information from the Grabs section, as a shot that takes more time can indicate the user had more difficulty with that particular shot. Also, it is useful to see if in the multiplayer part of the experience, the users spend the same time in a shot.
- **Shot success rate** - The percentage of shots that hit the target. Since the shots are categorized in 3 distinct types (made, close or missed) it can be studied not only the made shots, but also if they were close to it, or if they completely missed the target.
- **Distance to target using Time Control** - This is the distance of the ball to the basketball hoop in the moment the time was stopped. This allows to see the users behaviour regarding when is the moment they think is more suitable to achieve a successful shot.
- **Number of Signals** - The number of Signals the observer had to send to the manipulator in order to send a shot. This will allow to check if the communication was effective or not, as an increased number of Signals can be indicative of this.

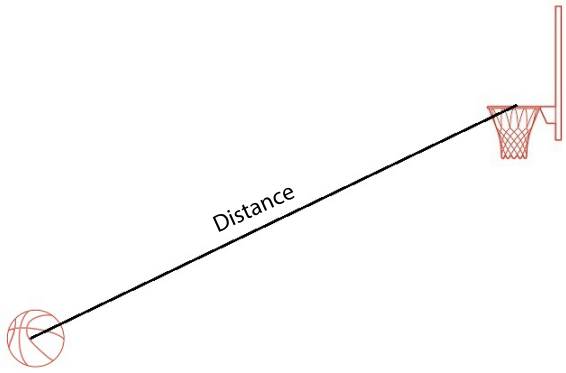


Figure 5.1: The distance parameter is the difference between the point at the center of the basketball and the center of the basketball hoop.[1]

Finally, by comparing all these results between them, it will be possible to establish relations between the effectiveness of the hologram manipulation vs the actual success of the users in fulfilling their objective.

5.4 User Studies

In this section, the gathered results from both methods presented in section 5.3 are detailed, beginning with a description of the user population that tested the prototype. Then, it is reviewed the single-player performance of the users as well as the multiplayer one, concerning collaboration between the users.

5.4.1 User Population overview

The tests were conducted with a total of 4 users, all invited through direct contact, and briefed on the subject of this thesis and what they would do in the experience. A note regarding the low number of participants and its consequences can be seen in section 6.1.

75% of the participants were male, and 25% were female. Concerning the participants' age, it has an average of 33.25, ranging from 28 to 39. About current experience with VR, 100% have had at least minimal experience with VR, but only 25% said they were frequent users. An overview of these results can be found in Tables 5.3, 5.4 and 5.5.

5.4.2 Single-player setting

The single-player part of the experience was focused on the hologram manipulation and remote interaction with the basketball. For this, the users must complete 3 tasks that total to 23 shots, as seen in section 5.1. All participants fulfilled these tasks in the same order. The data was retrieved after the application ended, and with the user questionnaires.

[1] Image adapted from https://www.dimensions.com/collection/basketball, last access June 2020

Table 5.3: Participants' Gender

Gender	# Participants
Female	1
Male	3

Table 5.4: Participants' Age

Age	# Participants
28-32	1
33-37	2
38-42	1

Table 5.5: Participants' experience with VR

VR Experience	# Participants
I've tried it a few times (beginner user)	2
I've used it sometimes (intermediate user)	1
I'm a frequent user (advanced user)	1

5.4.2.1 User questionnaires

The following questions were asked to each participant:

- **Sentence 1 (SP.1)** - "It was easy to control the hologram"
- **Sentence 2 (SP.2)** - "It was easy to adjust the direction of the ball"
- **Sentence 3 (SP.3)** - "It was easy to adjust the velocity of the ball"
- **Sentence 4 (SP.4)** - "Seeing the trajectory of the ball helped me to understand its movement"
- **Sentence 5 (SP.5)** - "The time control mechanism helped me improve my performance"

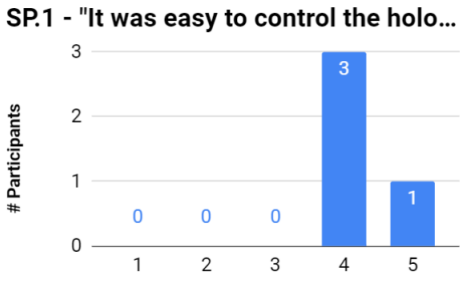
The first three questions were done to assess if the users could easily decouple both movements (rotation and scaling) available in the hologram, as well as their general liking on the system. SP.4 means to check if the trajectory was indeed a helper mechanism to the users, facilitating the usage of the hologram. SP.5 tries to verify if the participants thought that using the time control helped them to be more successful in their tasks, comparing to shots without time control usage.

Results The charts in Figures 5.2, 5.3 and 5.4 show the answers provided by the volunteers for the questionnaires given to them after the experience. For sentence SP.1, all participants indicated that the hologram was indeed easy to control, although only 25% did in fact considered to strongly agree to this statement.

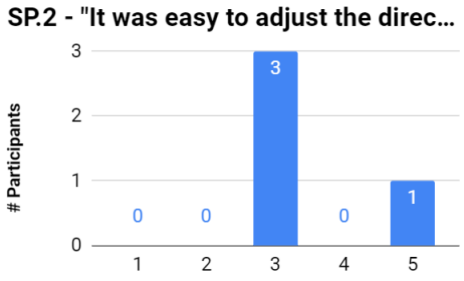
In sentence SP.2, with a mean of 3,5, it is ranked as the lowest in this category. 75% of the volunteers found that adjusting the direction of the ball was not very easy, although it wasn't hard as well.

Regarding sentence SP.3, contrary to the previous statement, users found the adjustment of the velocity to be simpler than the adjustment of the direction, with only 25% of users stating it was neither easy or hard.

Sentence SP.4 achieved the highest score possible, as 100% users indicated that seeing the trajectory strongly helped them to make the decision to where and how fast they should throw the ball.

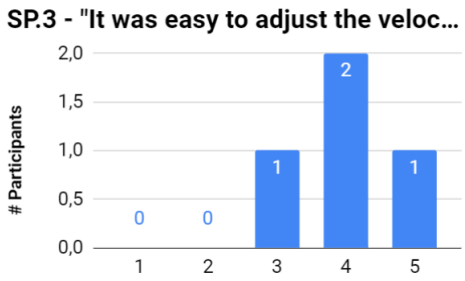


Mean	4,25
Standard Deviation	0,5
1st Quartile	4
Median	4
3rd Quartile	4,25

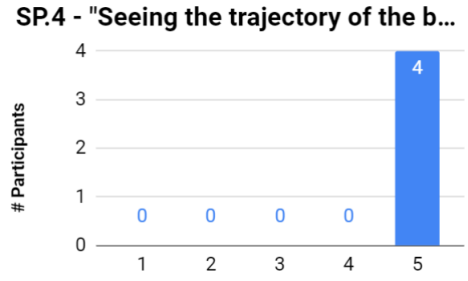


Mean	3,5
Standard Deviation	1
1st Quartile	3
Median	3
3rd Quartile	3,5

Figure 5.2: Single-player Questions SP.1 and SP.2



Mean	4
Standard Deviation	0,816
1st Quartile	3,75
Median	4
3rd Quartile	4,25



Mean	5
Standard Deviation	0
1st Quartile	5
Median	5
3rd Quartile	5

Figure 5.3: Single-player Questions SP.3 and SP.4

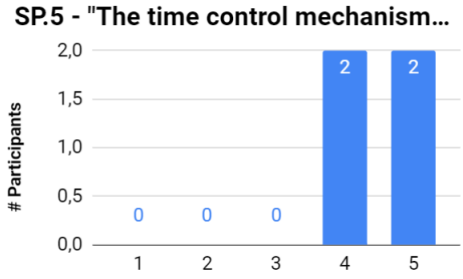
Finally, in sentence SP.5, all users scored the time control mechanism positively, and felt that it helped them improve their performance when comparing with regular shots.

An overview of these results can be seen in Table 5.6.

Table 5.6: Summary of the results of SP.1 to SP.5

	SP.1	**SP.2**	**SP.3**	**SP.4**	**SP.5**
Mean	4,25	3,5	4	5	4,5
Standard Deviation	0,5	1	0,816	0	0,577
1st Quartile	4	3	3,75	5	4
Median	4	3	4	5	4,5
3rd Quartile	4,25	3,5	4,25	5	5

Analysis The hologram manipulation paradigm proved to be an effective way to control an object remotely, giving a better foresight on how users perceive the movement that will come from



Mean	4,5
Standard Deviation	0,577
1st Quartile	4
Median	4,5
3rd Quartile	5

Figure 5.4: Single-player Question SP.5

the interaction. However, it comes with some downsides, specifically from what was assessed in sentence SP.2, as users said it was initially confusing to adjust the correct direction of the ball. This was one of the most challenging aspects in the creation of this prototype, and some measures were implemented to counter this difficulty, as explained in section 4.3 and reflected in Figures 4.7 and 4.9. Some users also gave some feedback regarding the control of the hologram at larger distances, suggesting some way to change the scale of movement at will. In fact, the larger the distance of the shot, the more unstable and difficult the shot will become, as a slight movement in the hologram's vector can lead to a very large difference in the outcome of the shot. This rate of success of the shot will be seen in more detail in the following sections regarding each of the experiment's tasks.

The users stated also that the trajectory was a great addition over a normal shot, and allows a greater sense of control over the object's manipulation. Having this trajectory allowed to clearly see where the object would land or hit, giving it a clear advantage over a direct manipulation paradigm. Although difficult, this trajectory could also be included in a direct manipulation setting, but with the disadvantage that it could only be seen from a frontal perspective, since the user would have the object in his hand. With this kind of interaction, the user can move around the court to see this trajectory from any perspective.

Additionally, the time control mechanism worked as it was intended, giving the users an advantage to use it instead of a regular shot, as they felt it increased their chances of success.

Finally, the System Usability Questionnaire scored an average of 76,875 points out of 100, which stands for a good usability score. The participants mainly referred that the system as whole was well integrated, that they felt confident while using the interaction paradigms of the experience and that they would like to use it again, while also stating that they did need to learn a number of different things before they felt comfortable using the system.

5.4.2.2 SP-DBS Task

As explained in section 5.1, the SP-DBS task consists in a series of shots grouped in 3 rows of increasing distance, with 4 shots per row. This is intended to not only be an easier introductory chapter to the experience, smoothing the difficulty curve to the user, but also to assess how the user will adapt to the increasing difficulty, and what will be the results of this.

Results The following charts show a summary of the retrieved results of the user tests on the whole population referred in section 5.4.1.

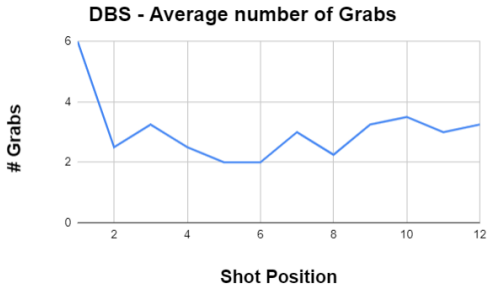
Regarding the number of grabs in each shot shown in Figure 5.5a, they were shown to be close in every participant, with only the first shot having higher number of grabs. The users made on average, 3,04 grabs, but if the first shot is not considered, this average goes down to 2,77. This is also true for the range in the number of grabs, which goes down to 6, not considering the first shot.

The chart presented in Figure 5.5b shows the average time spent in each shot. The first shot has a value of zero as this calculation is made with the difference between the time when the current shot has ended and the time when the previous shot ended. Since the first shot has no previous shot as reference, and also because the users took some more time to position themselves in the court and to get used to the hologram, it was counted as zero. In the first two rows of shots, the time spent per shot was similar, but it has increased in the last row of shots of higher difficulty, where the users took, on average 32,34 seconds per shot.

Concerning the number of shots hit, it is possible to see in Figure 5.6 that the total hit percentage is close to 50% , but greatly decreases in the last row of shots. The shots that were close to hit amount to exactly 50% of the total shots, while the ones that were completely missed represent 2,08% of the total shots taken.

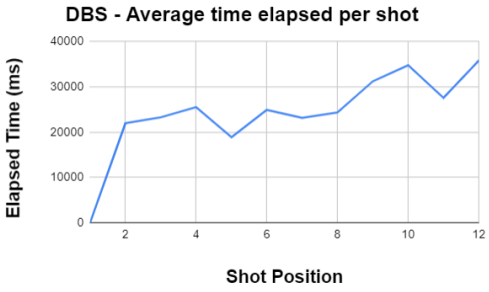
Analysis It is possible to observe that the average number of grabs has a very small difference between the easier shots and the harder ones. After the first shot, the number of grabs only slightly increases in the last row of shots, at a distance greater than $6m$. In fact, in the second row, of shots at intermediate distance, the number of grabs is actually smaller than the first row, indicating a quick adaptation to the hologram control. However, it is worth noting that the high value of grabs in the first shot derives from being the first ever shot taken after the freeplay session, where the user needs to move and position itself in an angle that gives him the best view to take a successful shot. Since it is also the user's first contact in the experience, it is the shot where the user is still learning and getting used to the hologram control setting.

The time taken per shot also strengthens the previous result, as it follows a similar path. All users took roughly the same amount of time in the first and second row of shots, increasing it the last one. This is due to the higher level of difficulty, as the users could be seen to take a more careful approach when adjusting the correct trajectory in the last row of shots. As they needed to send the ball with a greater velocity compared to the previous rows, this makes the shot more unstable, and more time was needed to adjust the hologram. It was also observed that the participants travelled a greater distance around the court in these shots, as they moved between the initial position of the ball to a position below the basketball hoop and to other lateral positions to



Mean	3,042
Standard Deviation	1,065
Range (Max - Min)	16

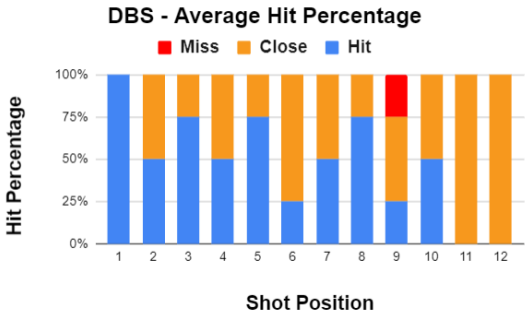
(a) Average number of Grabs per shot position



Mean	26487
Standard Deviation	5363,41
Range (Max - Min)	16952

(b) Average time spent per shot position

Figure 5.5: SP-DBS number of grabs and time spent per shot



Mean	47,92%
Standard Deviation	31,00%
Range (Max - Min)	100,00%

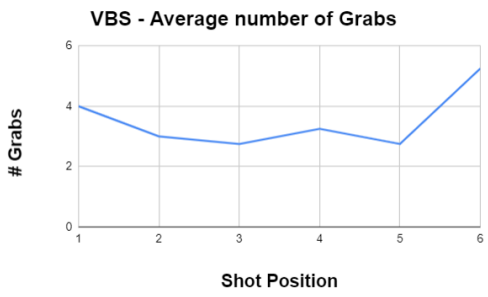
Figure 5.6: SP-DBS hit percentage

get a better perspective. Since the distance of the shot was larger, the distance the user travelled was also bigger, taking more time to complete.

As for the success rate, each row had a declining success percentage, of 68,8%, 56,3% and 18,8%, respectively. The progressively higher distance of the shots helps to explain this, as all the shots taken in the last row were taken at more than 6*m*, roughly around the 3-point line in the basketball court. It is still worth noting that only one shot of all the experiments has completely missed the target, while all others were either hit or were close to hit.

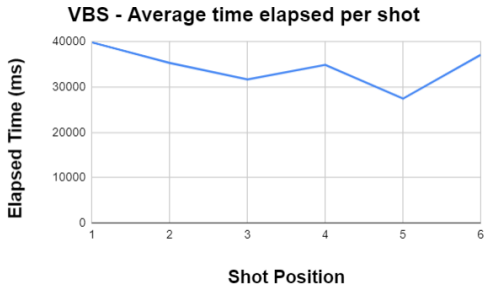
5.4.2.3 SP-VBS Task

The SP-VBS task was formed of 2 rows of 3 shots each, with the first being limited to a maximum velocity of 15 *m/s*, and the second to a minimum velocity of 15 *m/s*. These shots are taken from



Mean	3,500
Standard Deviation	0,975
Range (Max - Min)	10

(a) Average number of Grabs per shot position



Mean	34379
Standard Deviation	4329,57
Range (Max - Min)	49608

(b) Average time spent per shot position

Figure 5.7: SP-VBS number of grabs and time spent per shot

distances of intermediate difficulty, similar to the second row from the SP-DBS task. This task aims to deliver a different constraint to the users, forcing them to not only check the trajectory of the ball, but also the velocity the ball is thrown.

Results The number of grabs per shot increased in the second row of shots, with the first having an average of 3,25 grabs, with the second row having an average of 3,75 grabs per shot. This progression is shown in Figure 5.7a

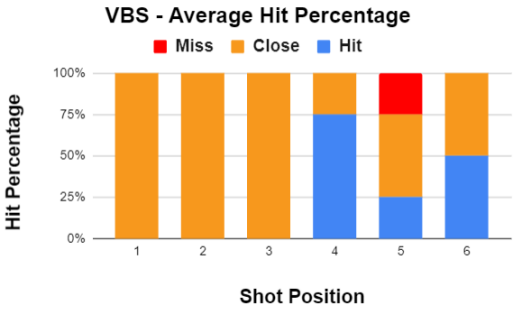
The average elapsed time per shot, shown in Figure 5.7b remained relatively constant in both rows, and ranged from a minimum average of 27,44 seconds to a maximum average of 39,85 seconds.

Regarding the shots made, present in Figure 5.8, the success rate in the first row of shots is null, but it considerably jumps to 50% in the second row of shots, while only completely missing one shot.

Analysis The number of grabs per shot increased significantly in comparison with the second row of shots from the SP-DBS task, considering they were taken from the same interval of distance. It has increased from an average of 2,31 grabs per shot to 3,5 grabs per shot. This is indicative of the increased challenge the users had, since they not only had to adjust the direction, but also take into consideration the velocity of the shot.

The time it took to complete the shots also significantly increased, registering a 29,8% increase when compared to the SP-DBS task. Still, when comparing both rows from the SP-VBS task, no significant difference in the time between shots can be observed.

Regarding the success rate, a staggering 0% success rate was observed in the first row of shots. During the observation of the experiments it was seen that the main cause of this low percentage was what is explained in section 4.5.2. Since the velocity had to be lower, the users were forced to make a lower parabolic trajectory, with the ball hitting the closer side of the rim almost every time, even though the trajectory showed that the ball would enter the hoop. However, in the second row of shots, the success percentage drastically increased to 50%, standing closer to what was observed



Mean	25,00%
Standard Deviation	31,62%
Range (Max - Min)	75,00%

Figure 5.8: SP-VBS hit percentage

in the second row of the SP-DBS task. These results are a clear indicator that the users have an obvious preference and feel way more comfortable using a greater parabolic trajectory rather than a more direct approach.

In fact, verifying the responses given to SP.3, seen in 5.4.2.1, it is possible to see that the majority of users felt more comfortable manipulating the velocity rather than the direction of the basket ball, and when this task restricted the velocity manipulation, it greatly affected the performance of the users.

5.4.2.4 SP-TCS Task

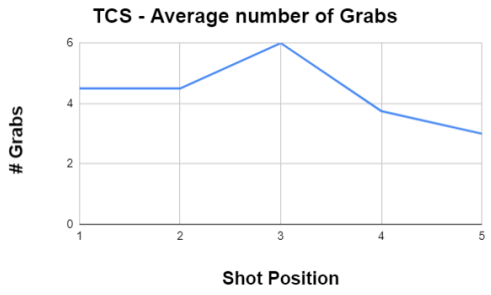
The SP-TCS task consists of a single row of 5 shots, from a distance similar to the the last row of shots of the SP-DBS task. This task aims to assess the effectiveness of the time control mechanic regarding hitting the target, as well as its ease of usage.

Results The average number of grabs per shot in this task is 4,35, as seen in Figure 5.9a, standing between an average of 3 and 6 grabs per shot, higher than any task previously seen.

The average time taken was of almost 50 seconds per shot, but is shown to be decreasing along the task (see Figure 5.9b. The time taken is considered since the beginning of this shot until the target is hit (or missed), consisting in the actions of pointing the shot, shooting, stopping the time, pointing again and shooting again from the position the ball is stopped, in this order.

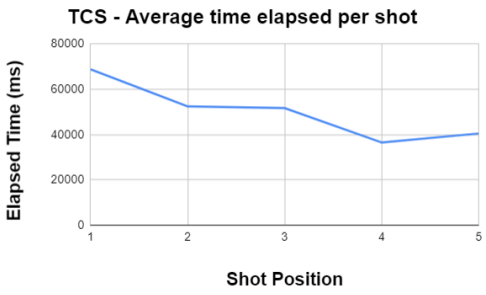
It is also relevant to know where the ball was stopped, in relation to the basketball hoop. This distance is calculated with the same method described in section 5.3.2. In Figure 5.9c it is possible to see that the average distance to the target when the ball is stopped is of 1,36m, and does not deviate much from this value, ranging even to values below 1m. Additionally, this distance continuously drops since the beginning of the task.

In relation to the success rate using the Time Control mechanism, it has the highest success rate among all other tests, standing at an average of 55%, with only one shot completely missed. Figure 5.10 shows these results.



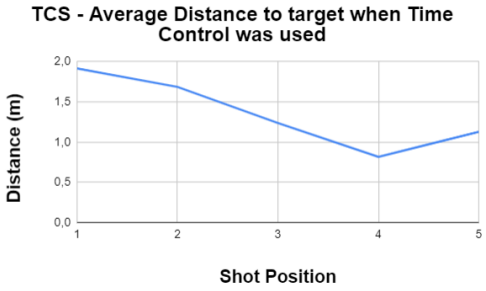
Mean	4,350
Standard Deviation	1,112
Range (Max - Min)	12

(a) Average number of Grabs per shot position



Mean	49985
Standard Deviation	12610,88
Range (Max - Min)	129331

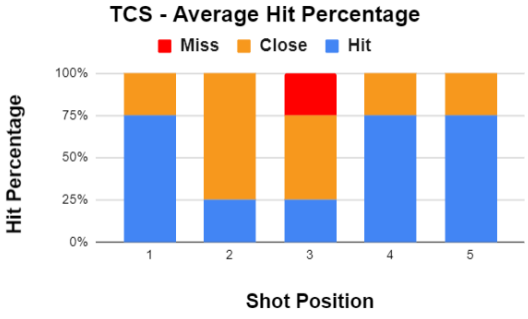
(b) Average time spent per shot position



Mean	1,36
Standard Deviation	0,44
Range (Max - Min)	4,39

(c) Average distance to target per shot position

Figure 5.9: SP-TCS number of grabs, time spent per shot and distance to target



Mean	55,00%
Standard Deviation	27,39%
Range (Max - Min)	50,00%

Figure 5.10: SP-TCS hit percentage

Analysis As seen, the average number of grabs per shot is the highest until this point. It is worth noting though, that each shot in this task is actually a combined shot, meaning that the user shoots the ball initially, stops the time, and then shoots it a second time. These values correspond to the total sum of these two *"sub-shots"*, since the users could not stop the time more than once. As such, this means that, on average, each of this "sub-shots" takes 2,175 grabs, which is the lowest value on this category. This behaviour was explained by observing that the users took more reckless first shots, without worrying much about hitting it right away. Most of them would just try to hit somewhere on the edge of the rim, or on the backboard and try to stop the time in that place, so they could easily score in their second attempt.

For the same reasons, the time took per shot was also increased, as pointing and taking two *"sub-shots"* took more time than a more direct approach. Still, it is seen that over the course of the task, the time per shot is decreasing, as the users would get more comfortable and skillful using this mechanic.

A very similar result happens in the distance to the target at the moment of usage of this mechanic. All users seemed to try to stop the time as close to the rim as possible, some even after hitting the rim or the backboard. This is a behaviour that maximizes their chances of hitting the target, as it turns the shot into one of minimal difficulty, since not only is the ball close to the rim, it already is at roughly the same height of it, or even higher.

As a result of these aspects, the success rate is high, standing above 50% of all shots taken. Even though these shots were taken from positions considered of a high difficulty level, using the Time Control mechanism turned these shots into a low difficulty shot, or, in the most extreme cases, of intermediate difficulty.

Relating this task with the results of Sentence SP.5 from the user questionnaires, it shows that the users did feel this increased level of success when using Time Control, contributing to an increased performance from them.

5.4.3 Collaborative setting

The multiplayer part of the experience was focused not only in the hologram manipulation, but also in the collaborative aspects of the experience, from the number and type of signals sent, to the success rate of this collaboration. The users completed 2 tasks with a total of 13 shots, which can be seen in section 5.1. All participants accomplished these tasks in the same order, with data being retrieved the same way it was done in the single-player setting.

5.4.3.1 User questionnaires

For the multiplayer part of the questionnaires, the following questions were made:

- **Sentence 1 (MP.1)** - "I was able to identify the presence of the other player easily"
- **Sentence 2 (MP.2)** - "It was easy to understand that the trajectory the other user was giving me was good or not"

- **Sentence 3 (MP.3)** - "I felt it was easier to score when interacting alone rather than with another player"
- **Sentence 4 (MP.4)** - "I felt my performance was better when interacting alone rather than with another player"
- **Sentence 5 (MP.5)** - "I felt it took more time to fulfill my tasks when interacting alone rather than with another player"
- **Sentence 6 (MP.6)** - "I felt the experience was more engaging when interacting alone rather than with another player"

The first question aims to verify if the users can quickly identify the presence of the other participant and his position without the need for communication. The second question is done to assess the effectiveness of the snapshot vectors that the manipulator will have in his hologram after receiving the signals from the observer. Questions MP.3 and MP.4 aim to verify the preference of the participant regarding his performance in a multiplayer setting or in a single-player. The objective of MP.5 is to see if the user does indeed have the impression of spending more time per shot in a multiplayer setting versus a single-player one. Lastly, MP.6 verifies if the users felt that the level of immersion of the experience was superior in multiplayer rather than single-player.

Results The Figures 5.11, 5.12 and 5.13 contain the charts representing the answers given to the multiplayer part of the questionnaire. Sentence MP.1 obtained a perfect score, as all participants said they could very easily identify the presence of the other user in the court.

The implemented communication mechanic of signal sending and snapshot vectors was also very well received, with a mean score of 4,25 in MP.2.

All the next questions tried to somehow compare the single-player experience with the multiplayer one. In MP.3, most users said they couldn't really tell if it was easier or harder to hit the target in either multiplayer or single-player, but 25% of the participants did prefer the single-player experience.

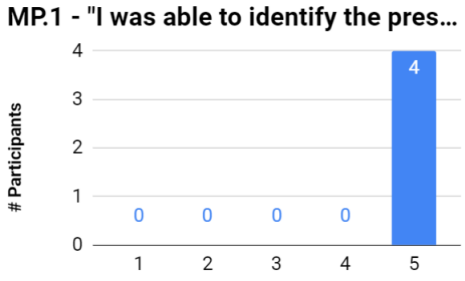
The same happens in sentence MP.4, with the same mean score of 2,5. The users felt that their performance in single-player mode was slightly better when compared to a multiplayer one.

In sentence MP.5 the answers also tended to slightly favor the single-player setting, as most users felt it took slightly less time to accomplish.

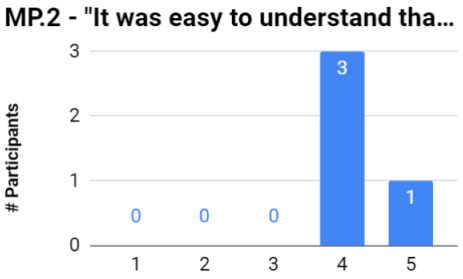
However, in MP.6, all users favored the multiplayer setting when it concerns to engagement and immersion in the experience.

An overview of these results can be seen in Table 5.7.

Analysis The non-verbal communication system that was implemented to complement verbal communication proved to be a good addition to the system as a whole, as was seen in MP.2. However, users did show some slight favor towards the single-player setting when concerning efficiency and performance, as they stated they felt it took less time and that it was slightly easier to score.

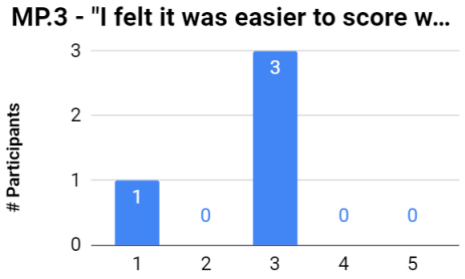


Mean	5
Standard Deviation	0
1st Quartile	5
Median	5
3rd Quartile	5

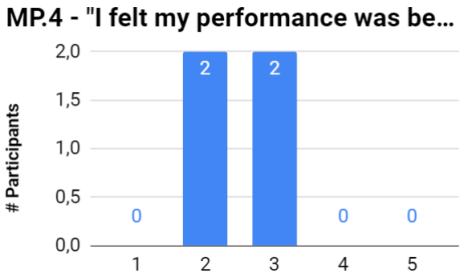


Mean	4,25
Standard Deviation	0,5
1st Quartile	4
Median	4
3rd Quartile	4,25

Figure 5.11: Multiplayer Questions MP.1 and MP.2

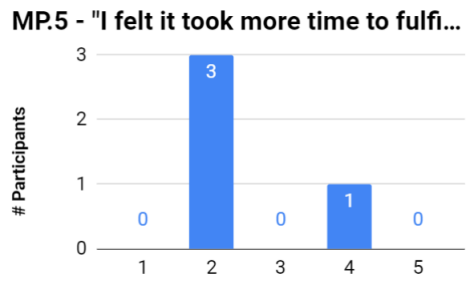


Mean	2,5
Standard Deviation	1
1st Quartile	2,5
Median	3
3rd Quartile	3

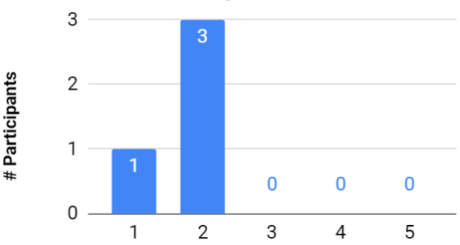


Mean	2,5
Standard Deviation	0,577
1st Quartile	2
Median	2,5
3rd Quartile	3

Figure 5.12: Multiplayer Questions MP.3 and MP.4



Mean	2,5
Standard Deviation	1
1st Quartile	2
Median	2
3rd Quartile	2,5



Mean	1,75
Standard Deviation	0,5
1st Quartile	1,75
Median	2
3rd Quartile	2

Figure 5.13: Multiplayer Questions MP.5 and MP.6

Table 5.7: Summary of the results of MP.1 to MP.6

	MP.1	MP.2	MP.3	MP.4	MP.5	MP.6
Mean	5	4,25	2,5	2,5	2,5	1,75
Standard Deviation	0	0,5	1	0,577	1	0,5
1st Quartile	5	4	2,5	2	2	1,75
Median	5	4	3	2,5	2	2
3rd Quartile	5	4,25	3	3	2,5	2

Regardless, even with a worse performance, users did prefer the multiplayer setting due to the immersiveness and engagement of that part of the experience. Both parts needed to communicate clearly and it was observed a clear evolution with the way they communicated in both multiplayer tasks. At first, the users serving as the manipulator were somewhat uncomfortable by not seeing the full trajectory of the ball, felt lost and some couldn't even correctly point their trajectory to the general direction of the basketball board. The verbal communication in this section was vaguer, with used sentences such as *"Should I point left?"* or even stating that they thought the trajectory was ok when the observer saw that it not only wasn't targeting the basketball hoop, it would go out of bounds. After the first couple of shots however, the communication between the two began to be more fluent, stating sentences such as *"I'm going to point upwards and with more force, so that it creates a larger arc"*. At later stages, they would also begin to state even precise values, such as increasing or decreasing 3 *m/s* to the current velocity. This quick improvement in the communication helped to achieve a greater sense of collaboration, leading to a better immersive factor.

5.4.3.2 MP-DBS Task

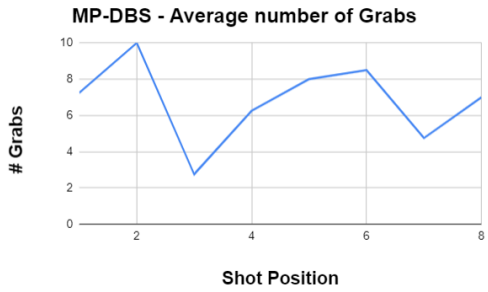
As it was seen in section 5.1, the MP-DBS task consists in a series of 2 rows of 4 shots each, of increasing distance. This task is intended to assess the communication effectiveness between the participants and how they help each other to reach their common goal.

Results The average number of grabs per shot has a mean of 6,813 and peaks at 10 grabs in the second position of the first row of shots, as seen in Figure 5.14a. The averages of each of the two rows are of 6,56 and 7,06, respectively, with the first one having a wider range than the second one.

The average time to take a shot is a little over a minute per shot, deviating 21,7 seconds from this value, as shown in Figure 5.14b.

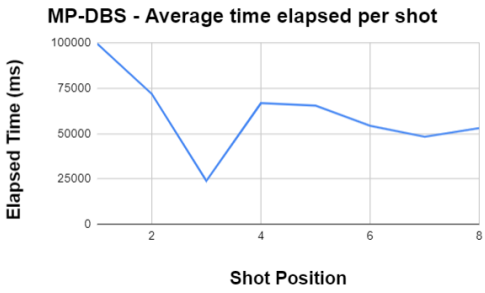
As for the success rate of this task, Figure 5.15 shows that an average of 28,13% of shots were hit, with the majority of them being close to hit.

Figure 5.16 shows the number of signals sent on each position, categorized by their type (red, yellow or green). The total average is of 5,25 signals per shot, and it stands for 5,5 in the first row of shots and 5 in the second row. The number of red and yellow signals is seen to be dropping since the beginning, while the green signals maintain constant during the task. The maximum number of signals sent in a single shot were 10.



Mean	6,813
Standard Deviation	2,259
Range (Max - Min)	29

(a) Average number of Grabs per shot position



Mean	60406
Standard Deviation	21742,37
Range (Max - Min)	302851

(b) Average time spent per shot position

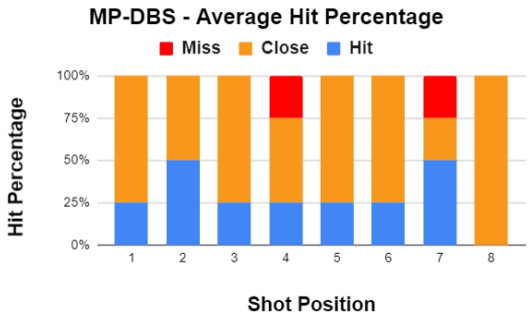
Figure 5.14: MP-DBS number of grabs and time spent per shot

Analysis The average number of grabs climbed substantially in this task when compared to the same task in a single-player setting, where it averaged 3,042 grabs. This means that the number of grabs more than doubled in a multiplayer setting. This can be explained due to the manipulator not being able to see a complete trajectory, but only a part of it, and having to rely on the observer's communication and on the received snapshot vectors to correctly align the trajectory.

The time spent in each shot also greatly increased in comparison to the SP-DBS task, from about 26 to 60 seconds, a 130% increase. Curiously, when users where asked about question **MP.5**, most said they only slightly felt this effect, saying they felt single-player took just a little less time to complete than multiplayer. This can be attributed to a greater immersive factor in a multiplayer setting, since both users were actively communicating with each other, which is confirmed by the answers given to **MP.6**. In this question, all participants referred that the experience was more engaging in a multiplayer setting.

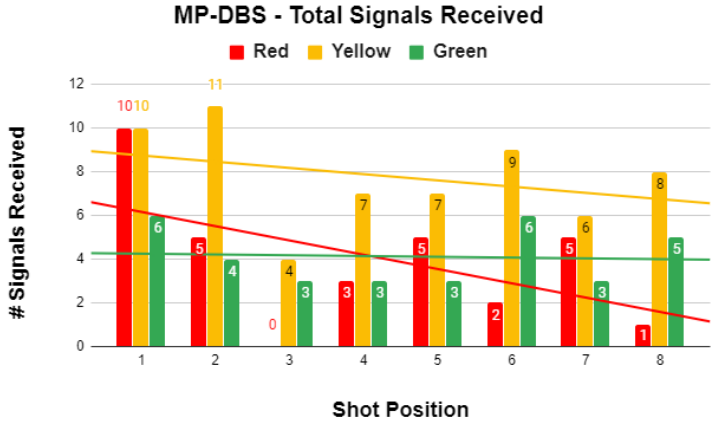
Regarding the success rate of this task, it stands as a mere 28,13%, compared to 37,5% of the SP-DBS task (if only considered the same last 8 shots), a 9.37% difference. The users seem to confirm this aspect in **MP.3** and **MP.4**, as both questions have a mean of 2,5 points. This suggests that this is an aspect that can be improved upon, as it currently positions itself as being a better tool for a single-player interaction, if we disregard the importance of presence and immersion in the experience.

The number of signals sent were greater in the first shots, but this can be explained due to being the first time both users were communicating, while also getting used to each others communication. The users also seem to get more accustomed to not seeing the full trajectory, as the number of yellow signals slightly decreased throughout the task and the red signals also decreased from an average of 4 in the first row of shots to 3,25 in the second row. In fact, **MP.2** seems to strengthen this conclusion, as all participants said that the signals sent by the observer were important for them to correctly calibrate the movement of the basket ball.



Mean	28,13%
Standard Deviation	16,02%
Range (Max - Min)	50,00%

Figure 5.15: MP-DBS hit percentage



Mean	5,25
Standard Deviation	2,89
Range (Max - Min)	11

Figure 5.16: MP-DBS number and type of signals received per shot position

5.4.3.3 MP-TCS Task

The MP-TCS task consists of a single row of 5 shots, just as the SP-TCS task, and are taken from a distance similar to this task. The objective of this task is to to assess the effectiveness of the time control mechanic in a collaborative environment. This adds a new, different challenge, as both users need to communicate twice in each shot: one for the regular shot, and another after the ball is stopped.

Results The average number of grabs per shot was recorded at 8,3 and it stands the highest number of the whole experience in this category. These values can be seen in Figure 5.17a.

This is also the case for the average time spent per shot, averaging almost 84 seconds per shot, with some of the recorded shots taking longer than longer than 2 minutes.

The distance at which the time control mechanic is used appears to remain relatively constant throughout this task, standing at an average of 0,98m, and deviating very slightly from this value, as shown in Figure 5.17c.

The evolution of the number of signals received also appears to decline over the time, with only the green signals remaining relatively constant. A mean of 5,07 were received during this task, as seen in Figure 5.19

Figure 5.18 shows that the success rate of the shots stands at exactly 40%, and that no shot was completely missed in this task.

Analysis The average number of grabs stands as the highest of the whole experience. However, just as it was seen in section 5.4.2.4, this task also can be divided into 2 *"sub-shots"*, as the users need to take another shot after stopping the ball. By linearly dividing the average between these 2 *"sub-shots"*, the average number of grabs per shot becomes 4,15, effectively becoming smaller when compared to the other multiplayer-based task, MP-DBS. This can be explained by two reasons: first, just as it happened in the SP-DBS task, the participants did not preoccupy in maximizing their hit chance right in their first attempt, but rather to attempt a *close* shot, so they could stop the time around the basketball hoop and attempt again from there. The second reason concerns the fact that since this is the second task in which the users collaborate, they achieve a greater sense of understanding each other, and become able to communicate more effectively. It was observed in this task that users would have more specific conversations such as *"Is 7 m/s too much?"*, *"Yes, probably around 5 m/s should do it"* instead of just trying things such as *"You should throw it upwards!"*. It was observed that this kind of interaction did in fact help the users to achieve their goal more effectively and rapidly.

In fact, the average time spent per shot in this task is only 23,55 seconds longer than the MP-DBS task, considering that in this one the user needs to stop the time and readjust his trajectory to a new shot. Still, when compared to the SP-TCS task, where the user also used the time control feature, it rose from 49,99 seconds to 83,96 seconds, a 67,97% increase. However, when compared to the previous task in section 5.4.3.2, where it rose 130% compared to the same task in the single-player setting, it represents a significant decrease in time, and proves a greater effectiveness in using both the hologram and the communication features as a whole.

The distance to the target in which the ball was stopped has also decreased in comparison to the SP-TCS task, from an average of 1,36m to 0,96m, which remarks a greater control over the usage of this mechanic with experience.

The number of shots hit also increased in comparison with the previous section, from 28,13% to 40%, and it is also worth noting that 0% of the shots were totally missed. This is a very good sign that, even though the participants were not able to fully achieve their objective, they were always very close to it. When compared to the same category of the single-player setting, it was verified a decrease of the success rate from 55% to 40%, which is similar to what happened in the previous section, where the multiplayer success rate also dropped.

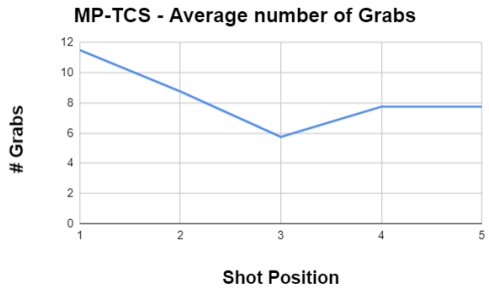
Regarding the communication aspects, there is a standout parameter that was observed. The MP-DBS obtained a mean of 5,25 signals per shot, however, in this task it has decreased to 5,07, even though this task consists of shots combined by two *"sub-shots"*, as seen before. The number of red and yellow signals is also shown to continuously decrease, and the the green ones to remain constant. Compared to the previous task, the red signals occupied a similar percentage of the total signals (23,3% in this task versus 21,8% in the MP-DBS task), the yellow signals decreased to 32,9% from 50,5% from the previous task, and the green signals increased to 43,8% from 27,7%. This difference is explained by the fact that since the second "sub-shot" is taken from a very close distance (and since the users were able to stop the ball closer than the SP-TCS task), even with just seeing 10% of the full trajectory they could already point the ball to the correct place, only requiring small adjustments to the trajectory. The increased percentage of green signals is explained by this second round of shots, since they doubled in amount, compared to the SP-TCS task.

5.5 Results Analysis and Discussion

The user studies conducted in the previous section helped to provide an insight over the effectiveness of a remote interaction method for certain types of tasks.

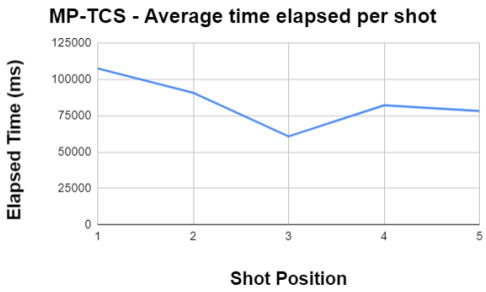
In the SP-DBS task, the users' first time facing this interaction paradigm, it was already shown to have good levels of success rate, even though it understandably declined when the distance and difficulty of the shots increased as well. When faced with a velocity restriction, the success rate took a toll as some users pointed out some concerns over managing the correct direction the ball would go. This is a concern that was identified in the implementation phase, as stated in section 4.3 regarding the hologram's axes and the guidelines along the edges of the court, and was countered with the said features. However, some more effort is surely needed to better deal with this problem, as it was observed that only with a larger amount of experience with the hologram's usage does this problem go away.

The time control feature proved to be a useful way to try out this paradigm, as it provides an advantageous way to interact with an object remotely, and while it obviously breaks the physics laws of the real world, it does so in a believable and conceivable way, not condoning the immersion



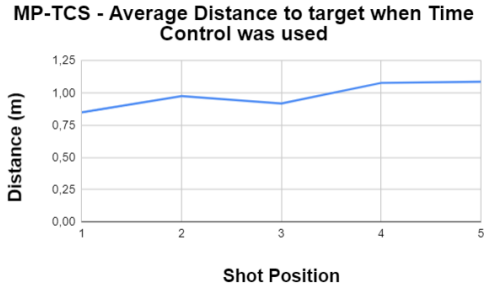
Mean	8,300
Standard Deviation	2,095
Range (Max - Min)	23

(a) Average number of Grabs per shot position



Mean	83958
Standard Deviation	17152,42
Range (Max - Min)	187157

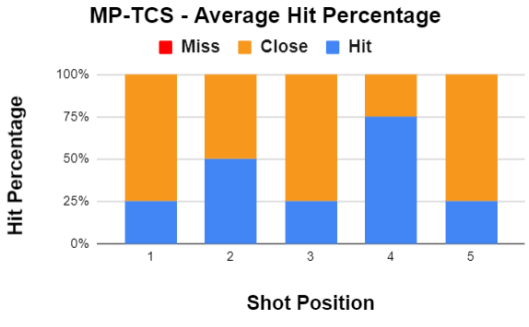
(b) Average time spent per shot position



Mean	0,98
Standard Deviation	0,10
Range (Max - Min)	0,95

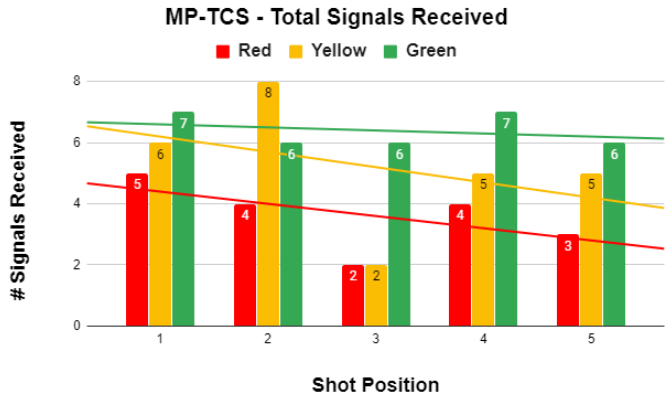
(c) Average distance to target per shot position

Figure 5.17: MP-TCS number of grabs, time spent per shot and distance to target



Mean	40,00%
Standard Deviation	22,36%
Range (Max - Min)	50,00%

Figure 5.18: MP-TCS hit percentage



Mean	5,07
Standard Deviation	1,79
Range (Max - Min)	6

Figure 5.19: MP-TCS number and type of signals received per shot position

of the overall experience. This type of interaction would be very hard to replicate via direct manipulation without breaking the sense of immersion.

The multiplayer section of the experience provided mixed results. On one hand, the success rate of the participants was lower, and took more time to complete when compared to the same tasks in a single-player setting. On the other hand, even with a lower percentage of success, users seem to prefer a multiplayer setting experience due to the elevated engagement and immersive factor, and having to constantly communicate with one another and cooperate to reach a common goal greatly contributes to this. Also, it is interesting to note that the curve of the average number of grabs and the one of the average time spent per shot are very similar in the multiplayer setting, but the same does not seem to happen as clearly in single-player. In multiplayer, the manipulator only appeared to grab the hologram's vector when it was told to, as he would move the vector, stop, wait for an indication of the observer and then move again. This is the opposite of what happened in the single-player mode, where the hologram's vector movement was more fluent and flowed more freely.

The test sessions took around 20 to 30 minutes each, and even in such a limited amount of time, the users already had much more control over the hologram at the end of the experience, showing a very quick adaptation to this remote interaction paradigm. The questionnaires at the end of the experience also reinforced what was seen in the experiment usage data, and provided valuable feedback from the experience.

Regarding a direct manipulation versus indirect remote manipulation paradigm, it was chosen to not include an evaluation of this aspect, as it didn't seem a fair comparison between two different methods of interaction. Directly throwing the ball at the basket can be more fun, but since it is still a weightless throw, without any kind of haptic feedback, it turns out to be very difficult to even have a shot considered as *"close"*, as seen in earlier internal tests. The indirect manipu-

lation implemented in this work allows for a finer tuning of both direction and velocity, allowing to predict the movement of an object before it actually happens. For these reasons, having a comparison between both types of manipulation would feel unfair, and since this indirect interaction mechanism would obviously provide much better results in this kind of experience, and as such, it wouldn't provide any kind of scientific breakthrough regarding object interaction. It is believed that both these interaction paradigms can coexist, even in the same setting or experience, and can be combined to offer a greater sense of immersion while at the same time, not losing control over an object and manipulate it exactly the same way the user is thinking.

5.6 Summary

This chapter began with the specification of the evaluation protocol used for this dissertation, specifying each individual task the users would perform in both single-player and multiplayer settings. The test session structure was then discussed, stating every step from the introduction to the experience, to filling the required questionnaires. It was later seen that the data that was used to evaluate this work was divided into two sections: the user questionnaires and the usage data that was automatically retrieved by the application. The user studies were then conducted, with each individual task stating their own results, analysis and comparison with other tasks, along with the results obtained from the questionnaires. Finally, a discussion over these results is presented, taking into account all the retrieved data as well as some factors that were observed during the experiences themselves.

Chapter 6

Conclusion and future work

Following the described in chapter 2, VR is a technology that includes multiple options since there are different equipment and techniques which require distinct configurations and influence the user interaction. As reviewed, immersion is a key factor in the user interaction and it is conditioned not only by the hardware but also by the way users can fully experience the VR scenario with a personal conscious inside the environment.

This consciousness is also influenced by the way this virtual reality respects the real world rules in terms of the physics laws, or disrespects those rules in a reasonable way that does not affect the immersion factor. There are already in use applications that take advantage of physics to represent and allow the interaction as is the case of VR simulators.

In the proposed solution, explained in the chapter 3, a remote object control interaction paradigm was developed to try to obtain a greater sense of control over an object, visualizing the consequences of said interaction before it would happen. In this case, a basketball scenario was the scene chosen to represent this paradigm, by representing the basket ball with a **hologram** attached to the user's hand. This hologram allowed the user to interact and analyze different behaviours according to the physics laws involved. In the end, the user was capable of manipulate the ball remotely and control its trajectory with the representation of the ball.

According the data collected from the users, this remote interaction allowed them to predict the ball trajectory while feeling engaged in the experience and achieving a reasonable success rate and an almost zero percent missed rate. This suggests that this novel interaction paradigm may be adequate to several scenarios.

Another important characteristic of this project was the **collaboration** between different users. According to the outcome of the questionnaires, this collaboration was considered to increase the engaging factor of the experience while executing the same kind of actions as the single player mode. Even though it did show that the users had a slightly smaller success rate in their objective, most seemed to prefer a multiplayer experience due to said immersion factor, reinforcing the importance of this aspect.

The platform was also developed in such a way that it could be customized in order to test different conditions for the same scenario, for example the wind and the gravity values.

6.1 Future Work

Due to the current Covid-19 pandemic situation, to the lockdown phase at the time of developing this thesis, and to health and hygiene concerns to all participants, it was not possible to bring more users to test the prototype to deliver more accurate results. A sample as small as the one used for this dissertation is not enough to correctly assess the effectiveness of such complex interaction methods, but it was done nevertheless, in order to establish a ground basis that it can be improved upon in the future, hopefully with more tests. In order to try and counter this effect, more system analytics were recorded per user, and were found to capably complement the results provided.

Some users provided insightful feedback regarding the usage of the hologram. It was verified that at greater distances, the success rate of the tool decreases, as the movements made to the hologram's vector become more unstable regarding the basket ball's final position. Some methods were implemented to smooth this additional difficulty, namely the stability factor, but it can still be improved upon. Changing the scale of the hologram would help to combat this, as increasing/decreasing the implemented scale of 1 unit to 10 *m/s* would impact the trajectory course.

The signaling system is another point that can be the aim for improvement. Instead of signaling if the trajectory is right/wrong/close, a more advanced system could be integrated, where the observer could send more detailed information, such as turning angles or velocity differences, that could eventually erase the need for the verbal communication, even though studies would need to be made regarding the impact it would have on the experience's immersion.

References

[AA13] Ferran Argelaguet and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. *Computers and Graphics*, 37(3):121–136, 2013.

[ABS+16] Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. Interaction Techniques Using Head Gaze for Virtual Reality. *2016 IEEE Region 10 Symposium (TENSYMP)*, pages 110–114, 2016.

[ADI17] Asmaa Saeed Alqahtani, Lamya Foaud Daghestani, and Lamiaa Fattouh Ibrahim. Environments and System Types of Virtual Reality Technology in STEM : A Survey. 8(6):77–89, 2017.

[AS02] Dana K Andersen and Richard M Satava. Virtual Reality Training Improves Operating Room Performance Results of a Randomized , Double-Blinded Study. 236(4):458–464, 2002.

[BC19] Costas Boletsis and Jarl Erik Cedergren. VR Locomotion in the New Era of Virtual Reality : An Empirical Comparison of VR Locomotion in the New Era of Virtual Reality : An Empirical Comparison of Prevalent Techniques. (April), 2019.

[BCK17] Costas Boletsis, Jarl Erik Cedergren, and Stian Kongsvik. HCI Research In Virtual Reality: A Discussion of Problem-Solving. pages 263–267, 2017.

[BH97] Doug A Bowman and Larry F Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*, 1997.

[BNS09] Anton Bogdanovych, Spanish National, and Simeon Simoff. Virtual Agents and 3D Virtual Worlds for Preserving and Simulating Cultures Virtual Agents and 3D Virtual Worlds for Preserving and Simulating Cultures. (September), 2009.

[Boa12] Yuri A G Vilas Boas. Overview of Virtual Reality Technologies. 2012.

[Bol17] Costas Boletsis. The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology. *Multimodal Technologies and Interaction*, 1(4):24, 2017.

[BRKD16] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. Point & Teleport Locomotion Technique for Virtual Reality. pages 205–216, 2016.

[BW18] Laurenz Berger and Katrin Wolf. WIM : Fast Locomotion in Virtual Reality with Spatial Orientation Gain & without Motion Sickness. 2018.

REFERENCES

[BWT01] Doug A Bowman, Chadwick A Wingrave, and Virginia Tech. Design and Evaluation of Menu Systems for Immersive Virtual Environments. (0106), 2001.

[Cox03] Christopher Cox. The use of computer graphics and virtual reality for visual impact assessments. 01 2003.

[DLDM12] Sébastien Delorme, Denis Laroche, Robert DiRaddo, and Rolando F. Del Maestro. Neurotouch: a physics-based virtual simulator for cranial microneurosurgery training. *Neurosurgery*, 71 1 Suppl Operative:32–42, 2012.

[DSZB05] Dedra Demaree, Stephen Stonebraker, Wenhui Zhao, and Lei Bao. Virtual reality in introductory physics laboratories. In *AIP Conference Proceedings*, volume 790, pages 93–96. AIP, 2005.

[Eic10] Christiane Eichenberg. *Application of 'Virtual Realities' in Psychotherapy: Possibilities, Limitations and Effectiveness*. 12 2010.

[FAR15] A A Faieza, A H Z Azreen, and M W Rohidatun. Virtual Reality Application in Design, Prototype and Test Drive Processes in Automotive Industry. 8354(4):24–32, 2015.

[FHZ96] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. Aperture Based Selection for Immersive Virtual Environments. *Proceedings of the 9th annual ACM symposium on User interface software and technology - UIST 96*, (Figure 1), 1996.

[FPB+16] Andrea Ferracani, Daniele Pezzatini, Jacopo Bianchini, Gianmarco Biscini, and Alberto Del Bimbo. Locomotion by Natural Gestures for Immersive Virtual Environments. pages 21–24, 2016.

[GBSC17] Elisa Gusai, Chiara Bassano, Fabio Solari, and Manuela Chessa. Interaction in an Immersive Collaborative Virtual Reality Environment: A Comparison Between Leap Motion and HTC Controllers. In *New Trends in Image Analysis and Processing – ICIAP 2017*, pages 290–300. Springer International Publishing, 2017.

[HWJ14] Alyssa Harris, Preston Tunnell Wilson, and Matthew Jackoski. Human Joystick : Wii-Leaning to Translate in Large Virtual Environments. pages 231–234, 2014.

[KB18] Yang Kuang and XiaoMei Bai. The research of virtual reality scene modeling based on unity 3d. pages 3 pp. –, 2018.

[Kit17] Comparing leaning-based motion cueing interfaces for virtual reality locomotion. *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, pages 73–82, 2017.

[KKP+00] Namgyu Kim, Gerard Jounghyun Kim, Chan-mo Park, Inseok Lee, and Sung H Lim. Multimodal Menu Presentation and Selection in Immersive Virtual Environments. pages 1–8, 2000.

[KLBL93] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[KLJK17] Mingyu Kim, Jiwon Lee, Changyu Jeon, and Jinmo Kim. A Study on Interaction of Gaze Pointer-Based User Interface in Mobile Virtual Reality Environment. 2017.

[KZFN16] Andreas Kunz, Markus Zank, Morten Fjeld, and Thomas Nescher. Real walking in virtual environments for factory planning and evaluation. *Procedia CIRP*, 44:257 – 262, 2016. 6th CIRP Conference on Assembly Technologies and Systems (CATS).

[LEB93] R. B. Loftin, M. Engleberg, and R. Benedetti. Applying virtual reality in education: A prototypical virtual physics laboratory. In *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*, pages 67–74, Oct 1993.

[LG94] Jiandong Liang and Mark Green. Jdcad: A highly interactive 3dmodeling system. *Computers and Graphics, 18(4):499–506*, 1994.

[LS18] Eike Langbehn and Frank Steinicke. Redirected Walking in Virtual Reality. (March), 2018.

[MBJS97] Mark R Mine, Frederick P Brooks Jr, and Carlo Sequin. Moving Objects In Space: Exploiting Proprioception In Virtual-Environment Interaction. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, USA, pp.19–26*, 1997.

[Mcl01] Hilary Mclellan. Virtual realities. 2001.

[MFW08] Eliana Medina, Ruth Fruland, and Suzanne Weghorst. VIRTUSPHERE: Walking in a human size VR "hamster ball". *Proceedings of the Human Factors And Ergonomics Society 52nd Annual Meeting*, (1991):2102–2106, 2008.

[Min95] Mark R Mine. Virtual Environment Interaction Techniques. *Technical report, UNC Chapel Hill*, 1995.

[MMSE17] M A Maksimov, V V Monakhov, H Sanchez, and J J Escobar. 2D and 3D virtual interactive laboratories of physics on Unity platform 2D and 3D virtual interactive laboratories of physics on Unity platform. pages 0–7, 2017.

[MNR18] João Moreira, Rui Nóbrega, and Rui Rodrigues. Mixed interactive/procedural content creation in immersive vr environments. 2018.

[MXM+15] Morgan Mccullough, Hong Xu, Joel Michelson, Matthew Jackoski, Wyatt Pease, William Cobb, William Kalescky, Joshua Ladd, and Betsy Williams. Myo Arm – Swinging to explore a VE. pages 107–113, 2015.

[NA19] Matthew E. Nelson and Benjamin Ahn. Virtual Reality Activities for Teaching Engineering Students Professional Development Skills. *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5, 2019.

[NSBK15] Mahdi Nabiyouni, Ayshwarya Saktheeswaran, Doug A. Bowman, and Ambika Karanth. Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. In *2015 IEEE Virtual Reality Conference, VR 2015*, pages 243–244. IEEE, 2015.

[OBKP18] Anne Helene Olivier, Julien Bruneau, Richard Kulpa, and Julien Pettre. Walking with Virtual People: Evaluation of Locomotion Interfaces in Dynamic Environments. *IEEE Transactions on Visualization and Computer Graphics*, 24(7):2251–2263, 2018.

[OC14] Murat Ozkok and Selcuk Cebi. A fuzzy based assessment method for comparison of ship launching methods. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 26:781–791, 03 2014.

[OH16] Antti Oulasvirta and Kasper Hornbæk. Hci research as problem-solving. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4956–4967. ACM, 2016.

[PBW96] Ivan Poupyrev, Mark Billinghurst, and Suzanne Weghorst. The Go-Go Interaction Technique : Non-linear Mapping for Direct Manipulation in VR. *UIST '96: proceedings of the 9th annual ACM symposium on user interface software and technology*, pages 79–80, 1996.

[Per19] Vasco Pereira. Virtual reality framework for collaborative applications. 2019.

[PFWM12] Tabitha C Peck, Henry Fuchs, Mary C Whitton, and Senior Member. The Design and Evaluation of a Large-Scale Real-Walking Locomotion Interface. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1053–1067, 2012.

[PMFW09] Tabitha C Peck, Student Member, Henry Fuchs, and Mary C Whitton. Evaluation of Reorientation Techniques and Distractors for Walking in Large Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):383–394, 2009.

[PNFG15] M.E. Portman, A. Natapov, and D. Fisher-Gewirtzman. To go where no man has gone before: Virtual reality in architecture, landscape architecture and environmental planning. *Computers, Environment and Urban Systems*, 54:376 – 384, 2015.

[Pri08] Colin Price. Learning physics with the unreal tournament engine. *Physics Education*, 43, 05 2008.

[RCM93] G Robertson, Stuart K Card, and Jock D Mackinlay. Three views of virtual reality: nonimmersive virtual reality. 1993.

[RE16] Juno Rae and Lizzie Edwards. Virtual reality at the british museum: What is the value of virtual reality environments for learning by children and young people, schools, and families. In *Proceedings of the Annual Conference of Museums and the Web, Los Angeles, CA, USA*, pages 6–9, 2016.

[RTBG17] Jose Luis Rubio-Tamayo, Manuel Gertrudix Barrio, and Francisco García García. Immersive Environments and Virtual Reality : Systematic Review and Advances in Communication , Interaction and Simulation. pages 1–20, 2017.

[SCP13] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual Reality on a WIM : Interactive Worlds in Miniature. (May), 2013.

[SGR+] Neal E. Seymour, Anthony G. Gallagher, Sanziana A. Roman, Michael K. O'Brien, Vipin K. Bansal, Dana K. Andersen, and Richard M. Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Ann Surg. 2002;236(4):458-463*.

[Sla14] Mel Slater. Grand challenges in virtual environments. *Frontiers in Robotics and AI*, 1, 05 2014.

REFERENCES

[SMM+10] C Savage, D McGrath, T McIntyre, M Wegener, and M Williamson. Teaching physics using virtual reality. *AIP Conference Proceedings*, 1263:126–129, 2010.

[SSV16] Mel Slater and Maria V. Sanchez-Vives. Enhancing Our Lives with Immersive Virtual Reality. 3(December):1–47, 2016.

[SVCL13] Frank Steinicke, Yon Visell, J Campos, and Anatole Lécuyer. Human Walking in Virtual Environments: Perception, Technology, and Applications. *Springer Verlag*, 2013.

[UAW+05] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks. Walking > walking-in-place > flying, in virtual environments. pages 359–364, 2005.

[Vil19] Jaime E. Villate. *Dinâmica e Sistemas Dinâmicos*. 5 edition, 2019.

[VSSK18] Sara Vlahovic, Mirko Suznjevic, and Lea Skorin-Kapov. Subjective Assessment of Different Locomotion Techniques in Virtual Reality Environments. In *2018 10th International Conference on Quality of Multimedia Experience, QoMEX 2018*, number c, pages 1–3. IEEE, 2018.

[WD17] Rustin Webster and Joseph Dues. System usability scale (sus): Oculus rift® dk2 and samsung gear vr®. 06 2017.

[WKM16] Preston Tunnell Wilson, William Kalescky, and Ansel Maclaughlin. VR locomotion : Walking > Walking in Place > Arm Swinging. 2016.

REFERENCES

Appendix A

Experiment Data Example


```
{
  "Distance": 0.181275889,
  "Velocity_vectorial": {
    "x": -2.21941018,
    "y": -3.126307,
    "z": 0.0545240529
  },
  "Velocity_magnitude": 3.8343904,
  "Shot_result": 4,
  "Timestamp": 63728704116335,
  "Time_since_last_shot": 29060,
  "Category": 8,
  "Grabs": [
    {
      "Timestamp_ini": 63728704096445,
      "Timestamp_fin": 63728704097131,
      "Velocity_vectorial_ini": {
        "x": -7.45708656,
        "y": 9.782446,
        "z": 6.041855
      },
      "Velocity_vectorial_fin": {
        "x": -8.811039,
        "y": 11.724968,
        "z": 7.273502
      },
      "Velocity_ini": 13.70432,
      "Velocity_fin": 16.3711052,
      "Time_grabbed": 686,
      "Velocity_difference": 2.66678524,
      "Velocity_difference_vectorial": {
        "x": -1.35395241,
        "y": 1.942522,
```

```
        "z": 1.231647
      },
      "Finished": true
    },
    {
      "Timestamp_ini": 63728704106431,
      "Timestamp_fin": 63728704108382,
      "Velocity_vectorial_ini": {
        "x": -8.811039,
        "y": 11.724968,
        "z": 7.273502
      },
      "Velocity_vectorial_fin": {
        "x": -2.22537947,
        "y": -3.12248,
        "z": 0.0188233964
      },
      "Velocity_ini": 16.3711052,
      "Velocity_fin": 3.8343904,
      "Time_grabbed": 1951,
      "Velocity_difference": -12.5367146,
      "Velocity_difference_vectorial": {
        "x": 6.5856595,
        "y": -14.8474483,
        "z": -7.25467825
      },
      "Finished": true
    }
  ],
  "Signals": [
    {
      "position": {
        "x": 6.37,
        "y": 0.1715005,
        "z": 7.38000059
      },
      "velocity_vectorial": {
        "x": -10.7331905,
        "y": 14.3425884,
        "z": 8.89318752
      },
      "velocity": 20.0,
      "signal_type": 2,
      "special_signal": ""
    },
    {
      "position": {
        "x": 0.1970203,
        "y": 3.42165542,
```

```
        "z": 12.4676132
      },
      "velocity_vectorial": {
        "x": -11.57634,
        "y": -16.3066711,
        "z": 0.284397
      },
      "velocity": 20.0000019,
      "signal_type": 1,
      "special_signal": ""
    }
  ],
  "TimeControlUsages": [
    {
      "Distance": 0.362271726,
      "Velocity_Ini": 16.3711052,
      "Velocity_vectorial_Ini": {
        "x": -8.811039,
        "y": 11.724968,
        "z": 7.273502
      },
      "Velocity_fin": 3.8343904,
      "Velocity_vectorial_fin": {
        "x": -2.21941018,
        "y": -3.126307,
        "z": 0.0545240529
      },
      "Finished": false
    }
  ]
}
```

Appendix B

Consent Form

DECLARAÇÃO DE CONSENTIMENTO

(Baseada na declaração de Helsínquia)

No âmbito da realização da dissertação de Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, intitulada **Physics-based platform for Collaborative Interaction in Virtual Reality**, realizada pelo estudante **Carlos Miguel Ferreira Lucas**, orientada pelo Prof. João Jacob e sob a co-orientação do Prof. Rui Nóbrega, eu abaixo assinado, _____, declaro que compreendi a explicação que me foi fornecida acerca do estudo que irei participar, nomeadamente o carácter voluntário dessa participação, tendo-me sido dada a oportunidade de fazer as perguntas que julguei necessárias.

Tomei conhecimento de que a informação ou explicação que me foi prestada versou os objectivos, os métodos, o eventual desconforto e a ausência de riscos para a minha saúde, e que será assegurada a máxima confidencialidade dos dados.

Explicaram-me, ainda, que poderei abandonar o estudo em qualquer momento, sem que daí advenham quaisquer desvantagens.

Por isso, consinto participar no estudo e na recolha de imagens necessárias, respondendo a todas as questões propostas.

Porto, _____ de Junho de 2020

_____

(Participante ou seu representante)

1/1