

# **POLITECNICO DI TORINO**

**Master's Degree in  
Cinema and Media Engineering**



**Master's Degree Thesis**

## **Non-verbal communication in Social VR: Exploring new ways to handshake**

**Supervisors**

Prof. Fabrizio LAMBERTI

Prof. Adalberto SIMEONE

**Candidate**

Irene CHECO

**December 2022**



# Abstract

Technology has drastically changed the way people communicate with each other: talking and reaching out to someone has become easier, even across great distances, thanks to smartphones and computers. Social networks, later, helped us to expand this even further, allowing us to meet new people from all over the world, share experiences and bond with them from the comfort of our homes.

With the advent of Virtual Reality (VR), people felt the need to make online socializing experience more “physical” and “real” thanks to the immersive nature of this technology. This gave way to a new world of communication: Social VR platforms. Here users can create their avatar, customizing it as they prefer and meet old and new friends in a virtual environment (also customizable), where they can play games together, hang out and even attend events thanks to the streaming options.

Although Social VR already made the experience quite immersive for the user, these platforms are still lacking when it comes to non-verbal communication.

Literature shows that non-verbal communication is usually perceived by the users as a positive experience because it makes the interactions more immersive and similar to offline face-to-face interactions, helping to initiate the conversation also with strangers.

It is, therefore, an important component for the immersion of the user that warrants more research attention.

The aim of this thesis work is to improve the range of expressivity in non-verbal communication, designing and developing some new interaction options, by building on previous studies on how non-verbal communication is perceived by users.

The outcome of the work was, then, tested on target users to understand how these new interactions affect their experience of Social VR and how they could be integrated in the currently available platforms.



# Table of Contents

<b>List of Figures</b>	VII
<b>Acronyms</b>	IX
<b>1 Introduction</b>	1
1.1 The impact of social network websites on our life . . . . .	1
1.2 The perks of immersive virtual reality . . . . .	2
1.3 A new frontier for social interaction: Social VR . . . . .	4
1.4 Non-verbal communication . . . . .	5
<b>2 State of the art</b>	8
2.1 Virtual reality . . . . .	8
2.1.1 Origins . . . . .	8
2.1.2 Input devices . . . . .	15
2.1.3 Output devices . . . . .	16
2.2 Social VR platforms . . . . .	16
2.2.1 VRChat . . . . .	17
2.2.2 AltSpaceVR . . . . .	17
2.2.3 RecRoom . . . . .	18
2.2.4 Mozilla Hubs . . . . .	18
2.3 Non-verbal communication in Social VR platforms . . . . .	20
2.3.1 Talking without a voice . . . . .	20
2.3.2 Social VR platforms group walk-through . . . . .	21
2.3.3 How do I make this thing smile? . . . . .	22
2.4 Examples of case studies . . . . .	23
2.4.1 Pose estimation with gloves and trackers . . . . .	24
2.4.2 Pose estimation with optical camera and machine learning . . . . .	25
2.4.3 NVC classification and their presence in Social VR existing plat- forms . . . . .	26

<b>3</b>	<b>Technologies</b>	28
3.1	Hardware	28
3.1.1	Oculus Quest 2	29
3.2	Software	31
3.2.1	Unity 3D	31
3.2.2	XR Interaction Toolkit Unity plug-in	32
3.2.3	Photon Pun 2 Unity package	33
3.2.4	Visual Studio	33
3.2.5	Blender	34
3.2.6	Substance Painter	35
<b>4</b>	<b>Design</b>	37
4.1	Design	37
4.1.1	Handshake Options	38
4.1.2	Test scenario	41
<b>5</b>	<b>Development</b>	42
5.1	Base VR scene setup	42
5.2	Multiplayer setup	42
5.2.1	Server connection	43
5.2.2	Joining rooms and spawning players	43
5.2.3	Players setup	44
5.3	Handshake: Option 1	45
5.3.1	Player features	45
5.3.2	Handshake mechanism	45
5.4	Handshake: Option 2	47
5.4.1	Player features	47
5.4.2	Handshake mechanism	47
5.5	Handshake: Option 3	49
5.5.1	Player features	49
5.5.2	Handshake mechanism	49
5.6	Handshake: Option 4	51
5.6.1	Player features	51
5.6.2	Handshake mechanism	51
5.7	Test setup	52
5.7.1	Virtual environment	52
5.7.2	NPC	53
5.7.3	Tutorial scenes	55

<b>6</b>	<b>User study and results</b>	57
6.1	Experimental protocol . . . . .	57
6.1.1	Pre-experience questionnaire . . . . .	58
6.1.2	Post-variant questionnaire . . . . .	59
6.1.3	Post-experience questionnaire . . . . .	59
6.2	Results analysis . . . . .	60
6.2.1	Participants demographics . . . . .	60
6.2.2	Options comparison . . . . .	61
<b>7</b>	<b>Conclusion and future work</b>	64
<b>A</b>	<b>Pre-experience quetionnaire</b>	65
<b>B</b>	<b>Post-variant quetionnaire</b>	66
<b>C</b>	<b>Post-experience quetionnaire</b>	72
	<b>Bibliography</b>	74
	<b>Aknowledgements</b>	77

# List of Figures

1.1	a) Non-immersive VR ( <i>url: <a href="https://rb.gy/ey34ue">https://rb.gy/ey34ue</a></i> ) b) Semi-immersive VR( <i>url: <a href="https://rb.gy/5pxevw">https://rb.gy/5pxevw</a></i> ) . . . . .	3
1.2	Immersive VR ( <i>url: <a href="https://rb.gy/8po3tp">https://rb.gy/8po3tp</a></i> ) . . . . .	4
1.3	Social VR platform ( <i>url: <a href="https://rb.gy/lq8hwj">https://rb.gy/lq8hwj</a></i> ) . . . . .	5
2.1	Pygmalion’s Spectacles ( <i>url: <a href="https://rb.gy/bcb16j">https://rb.gy/bcb16j</a></i> ) . . . . .	9
2.2	Sensorama ( <i>url: <a href="https://rb.gy/zbxmte">https://rb.gy/zbxmte</a></i> ) . . . . .	9
2.3	The Sword of Damocles ( <i>url: <a href="https://rb.gy/qkpw7y">https://rb.gy/qkpw7y</a></i> ) . . . . .	10
2.4	a) Virtuality arcade machines ( <i>url: <a href="https://rb.gy/1n3ftl">https://rb.gy/1n3ftl</a></i> ) b) An exmple of a CAVE room( <i>url: <a href="https://rb.gy/3sjcx6">https://rb.gy/3sjcx6</a></i> ) . . . . .	12
2.5	a) Sega VR ( <i>url: <a href="https://rb.gy/sltesc">https://rb.gy/sltesc</a></i> ) b) Virtual Boy ( <i>url: <a href="https://rb.gy/vllwz1">https://rb.gy/vllwz1</a></i> )	13
2.6	a) Oculus Rift ( <i>url: <a href="https://rb.gy/ivn8gg">https://rb.gy/ivn8gg</a></i> ) b) Google Cardboard ( <i>url: <a href="https://rb.gy/8cbrgu">https://rb.gy/8cbrgu</a></i> ) . . . . .	13
2.7	PlayStation VR ( <i>url: <a href="https://rb.gy/65ziuv">https://rb.gy/65ziuv</a></i> ) . . . . .	14
2.8	Examples of types of observed non-verbal behaviors [4] . . . . .	20
2.9	Set of tasks considered and the interaction cycle they use in each platform [3] . . . . .	22
2.10	Inventory [6] . . . . .	23
2.11	SVR system architecture incorporating non-verbal cues [12] . . . . .	25
2.12	OpenPose human pose detection with image (left), without image (right) [13] . . . . .	25
2.13	Social VR platforms analysis (March 2022) . . . . .	27
3.1	Headsets comparison ( <i>data from url: <a href="https://rb.gy/kxvbby">https://rb.gy/kxvbby</a></i> ) . . . . .	29
3.2	Oculus Quest 2 ( <i>url: <a href="https://rb.gy/9weoj8">https://rb.gy/9weoj8</a></i> ) . . . . .	30
3.3	Oculus Quest 2 controllers ( <i>url: <a href="https://rb.gy/ddzsjs">https://rb.gy/ddzsjs</a></i> ) . . . . .	31
3.4	Unity interface . . . . .	32
3.5	Blender interface . . . . .	35
3.6	Substance Painter interface . . . . .	36



4.1	In order: VRChat Action Menu, RecRoom Espresso Menu, Mozilla Hubs Emoji Spawners, AltSpaceVR Radial Menu . . . . .	39
4.2	Sketch of the environment for the test scenario . . . . .	41
5.1	a) Starting panel b) Waiting panel c) Confirm panel . . . . .	46
5.2	a) Starting panel b) Waiting panel c) Confirm panel . . . . .	48
5.3	a) Starting panel b) Waiting panel c) Confirm panel . . . . .	50
5.4	a) Starting panel b) Confirm panel c) Shaking hands . . . . .	52
5.5	a) Town Hall b) Cafeteria . . . . .	53
5.6	a) Grocery store b) Gym . . . . .	53
5.7	Mayor . . . . .	54
5.8	Waitress . . . . .	55
5.9	Tutorial scene . . . . .	56
6.1	Participants demographic data . . . . .	61
B.1	* questions repeated for each variant. . . . .	67
B.2	* questions repeated for each variant. . . . .	69

# Acronyms

**2D** Bi-dimensional

**3D** Three-dimensional

**API** Application Programming Interface

**CAVE** Cave Automatic Virtual Environment

**CPU** Central Processing Unit

**DLL** Dynamic Link Library

**DoF** Degrees of Freedom

**FOV** Field Of View

**GPL** General Public License

**GUI** Graphical User Interface

**HCI** Human-Computer Interaction

**HMD** Head-Mounted Display

**IDE** Integrated Development Environment

**IPD** Interpupillary Distance

**LCD** Liquid Crystal Display

**NPC** Non-Player Character or Non-Playable Character

**NVC** Non-Verbal Communication

**RAM** Random Access Memory

**RPC** Remote Procedure Calls

**SNS** Social Network Sites

**UI** User Interface

**VR** Virtual Reality

**XR** Extended Reality



# Chapter 1

## Introduction

With the evolution of technology, it has gone hand in hand a rapid evolution of the way of communication used by people, increasingly characterized by immediacy and independence from time and space. Over the past few years, the spread of virtual reality devices led research to try to combine the immediacy provided by the digital and the sense of presence and immersion offered by Virtual Reality (VR), in order to make digital communication more and more similar to face-to-face communication.

There is, nevertheless, a dimension of communication that has not been sufficiently explored in these technologies yet: non-verbal communication, which is what mainly distinguishes real-life meetings.

### 1.1 The impact of social network websites on our life

The Greek philosopher Aristotle defines the human as a social animal, incapable of living alone. The evolution of technology and communication helped us fulfill this inborn need of relationships: starting from telephones, then through internet and emails, we became able to talk and reach out to someone, even when far apart.

It was the advent of social network sites, though, that really changed our way of socializing.

Nicole B. Ellison defines social network sites as a web-based services that allows people to build a profile (public or semi-public), make a list of users with whom they

share a connection and navigate through this list and other people's lists [1]. Depending on the platform, they sometimes encourage people to share pictures, videos, stories, attracting different audiences based on common passions or activities.

This structure allows users to make their network and their interests visible and, sometimes, it could result in connections between individuals that would not otherwise be made. This opportunity to start new relationships with strangers is what differentiates SNS from the other technologies mentioned above.

Thanks to these new medias, communication and interaction can now take place online, without having to face each other at the same place and same time: internet has become a place to meet, interact, keep in touch with old friends and form new relationships [2].

## 1.2 The perks of immersive virtual reality

The definition of virtual reality comes from the combination of the words "virtual" and "reality": it is, in fact, the building of an artificial world, generated by computer, in which the user is placed. This technology comes from the union of three closely linked components: interaction, presence and immersion.

Interaction means how the user gets in touch and interacts with the virtual environment: it requires a real-time simulation and the engagement of all human sensory canals (sight, sound and touch).

Presence is the embodiment, the sense of belonging to the artificial world. The environment has to be designed to maximize this feeling and the user has to be willing to be a part of this environment (suspension of disbelief).

Immersion occurs when the user perceives only the stimuli from the virtual world and loses touch with the physical reality: the sensory feedbacks need to be consistent with what happens in the artificial environment.

There are three types of virtual reality:

- *Non-immersive VR* (Desktop VR or Windows on Worlds, WoW): made up by a monitor and a few input devices (such as mouse, controller and keyboard). It is very limited when it comes to immersivity and complexity of the interaction, however, it is the most affordable in terms of costs and, therefore, the most common. An example could be video gaming consoles, as shown in Fig. 1.1.

- *Semi-immersive VR*: it provides a partially virtual environment. This kind of virtual reality is often used for training and educational purposes and relies on high-resolution displays or projectors and powerful computers. This technology is used, for example, for flight simulators. Another example is provided in Fig. 1.2
- *Immersive VR*: it is the virtual reality properly so called, in which all the components that characterize this technology are present (presence, immersion and interaction). It may consist in different input and output devices, although the most common are the head-mounted displays as output devices and the controllers as input devices.

Since almost the entire human field of view is covered, the perspective changes according to the head movements and the stereoscopic vision is provided, this technology allow the user to be completely immersed in the virtual environment, increasing his engagement and the realism of the experience. An example of a VR headset is shown in Fig. 1.3.

With the advancement of technology, immersive VR devices are quickly spreading through the users, pushing the companies to develop an ever-increasing number of ad hoc platforms and applications.



Figure 1.1: a) Non-immersive VR ([url: https://rb.gy/ey34ue](https://rb.gy/ey34ue))  
b) Semi-immersive VR([url: https://rb.gy/5pxevw](https://rb.gy/5pxevw))



Figure 1.2: Immersive VR  
(url: <https://rb.gy/8po3tp>)

### 1.3 A new frontier for social interaction: Social VR

The spread of immersive VR led the companies to try to transfer the social dimension, typical of social network websites, to these virtual environments. These artificial worlds, in fact, allow you to create an experience that is more and more similar to a face-to-face interaction in real life, thanks to the level of immersion that they provide to the users [3].

This is how Social VR platforms were born: virtual spaces that users can access through VR head-mounted displays and where they can interact with each other, communicating via their avatars. Moreover, the tracking of the head and hands movements, provided by the immersive VR technology, allow to broaden the spectrum of communication modes available to users, adding non-verbal interaction to verbal communication.

The developing of Social VR has drawn the attention of research to issues, such as the design of these platforms and the ways of communication and interaction of the users. However, just a few explored the using of non-verbal communication and the impact that this has on interactions inside these platforms [4]. An example of of these platforms is shown in Fig. 1.4.



Figure 1.3: Social VR platform  
(url: <https://rb.gy/lq8hwj>)

## 1.4 Non-verbal communication

Wang, in her study on non-verbal communication's effect on interpersonal communication [5], highlights the importance of NVC (non-verbal communication) in social interactions quoting Larry A. Samovar: "most experts think that in face-to-face interpersonal communication, only about 35% of the *social meanings* is expressed by verbal behaviors, the rest is expressed by nonverbal communication". But what is meant by non-verbal communication?

Maloney et al.[4] define non-verbal communication as all those types of communication which convey information without a purely linguistic content. Given the breadth of this category of communication, literature has provided over time several classifications. In summary, we can identify four main groups:

- *Facial behavior*: facial expressions can communicate universal information about a person's state, his/her intentions and emotions. They can be intentional, as a smile, or unconscious, as blushing or a reaction of disgust.
- *Gaze*: together with facial expressions, they communicate a person's intentions and feelings, but depending on when that person is looking, one can also understand the object of his/her attention.
- *Gestural behavior*: it includes the movement of hands and other parts of the body. It highlights the concepts expressed verbally and it also replaces speech when it is ineffective or incomprehensible. However, gestures could have very different



meanings across different cultures, becoming, sometimes, more of an obstacle than a support.

- *Spatial behavior*: also called *proxemics* [6], it means the distance between two people interacting. Individuals use it to show comfort, unease, fear, anger, etc. In literature, four zones have been identified [7]: intimate, personal, social and public. Each zone is characterized by a distance range and is used for different types of communication.

In summary, the importance of NVC in interpersonal interactions is widely recognized. Nevertheless, with the development of the digital and of computer-mediated communications, where a physical body is absent, this dimension has gone missing.

In the case of non-immersive VR, they tried to compensate this lack introducing symbols and emoticons or, with avatar-mediated communication, also using avatar's pre-defined and scripted behaviors and movements.

Immersive VR, instead, allows to broaden the spectrum of NVC available to the user, including a wider gestural expressiveness directly controlled by the person (not pre-defined), more control on spatial behavior and, sometimes, on gaze (by eye-tracking).

Furthermore, non-verbal communication could include touching, considering that tactile communication is probably the first form of communication the human knows since his birth. In the book *Nonverbal Communication in Human Interaction* [8], Knapp et al. provide several classifications of touching behavior: based on types of touching (such as handshake, pat, hug, kiss, etc.), based on the intimacy suggested by that touching (professional, friendship, love, etc.) and based on the meaning and impact of touching (play, influence, healing, etc.). The integration of this kind of interaction in virtual worlds is more complex than the previous, since it requires not just the presence of a physical body, but also the possibility to physically touch the other person.

To fill this gap, research is focusing on developing new input devices, more accurate and able to give tactile feedback to the user (e.g., haptic gloves).

This thesis, instead, aims to analyze the current state of NVC in social virtual reality (including touching) and to explore and develop a few proposals of interactions that could be integrated in existing platforms, without the need of devices other than VR headsets' controllers. In particular, this project will focus on handshake, since it is the first gesture that opens a new acquaintance and a natural interaction used in many

social contexts as a symbol of greeting, farewell and congratulations.

# Chapter 2

## State of the art

In this chapter the history of virtual reality will be briefly covered, with its developments and the devices currently available. Afterwards, the Social VR platforms taken into examination for this project will be outlined.

### 2.1 Virtual reality

The struggle of finding a definition of virtual reality appropriate to every facet of this concept affects also the ability to determine its exact origins. In this case, the definition provided by the Cambridge Dictionary [9] will be used: *"a set of images and sounds, produced by a computer, that seem to represent a place or a situation that a person can take part in"*.

Therefore, in this analysis only the cases where it is necessary to have an external device in support of the experience will be included.

#### 2.1.1 Origins

The history of VR began about a century ago, with the science fiction *Pygmalion's Spectacles* (1935) by Stanley G. Weinbaum, shown in Fig. 2.1. This story, in fact, shows the idea of a pair of goggles that lets who wears them experience a fictional world, *"a movie that gives one sight and sound [...] taste, smell, and touch. [...] You are in the story, you speak to the shadows (characters) and they reply, and instead of being on a screen, the story is all about you, and you are in it"*.

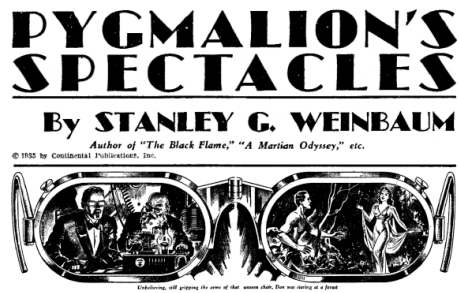


Figure 2.1: Pygmalion's Spectacles  
(url: <https://rb.gy/bcb16j>)

The first physical device has been tested thanks to the film industry: the director Morton Heilig, in 1955 in his essay *The Cinema of the Future*, theorized an *Experience Theatre*, with the aim of making people feel as if they were in the movie with an experience able to involve all the sensory channels realistically. He, then, built up a prototype in 1962, the *Sensorama*, shown in Fig. 2.2, with five movies that could be projected by this device. It was a mechanical device (since it had been built before digital computers) with a structure similar to an old Arcade cabinet, equipped with stereophonic speakers, a stereoscopic display, fans, smell generators and a vibrating chair. The project failed, though, because of a lack of founding.



Figure 2.2: Sensorama  
(url: <https://rb.gy/zbxmte>)

In 1960, then, Heilig patented a device for individual use, called *The Telesphere Mask*, which provided stereoscopic vision and stereophonic sound and it was able to stimulate also olfactory sensations. It showed a lot of similarities with later Head-Mounted Displays (HMD), but it did not have any motion tracking. The following year, two engineers from Philco Corporation (Comeau and Bryan) developed the *Headsight*, which featured a screen for each eye and a magnetic tracking system, linked with a closed-circuit camera: head movements, therefore, moved a remote camera, allowing the user to realistically look around the environment. This device was developed to let the military immersively monitor dangerous situations remotely. *Headsight* has been the first step in the evolution of VR head-mounted displays.

Ivan Sutherland introduces, in 1965, the concept of *Ultimate Display*, a device able to simulate reality to the point that you could not tell the difference from the actual reality. A computer would create the virtual world and maintain it in real time; the user could see it through an HMD and could realistically interact with virtual objects, also thanks to sound and tactile feedbacks.

In 1968, then, with his student Sproull, Sutherland created the first VR head-mounted display connected to a computer instead of a camera, *The Sword of Damocles*, shown in Fig. 2.3. It was a big device hanging from the ceiling by a mechanical arm (feature that gives it the name) to facilitate the users' movements, since the system was too heavy to be comfortably worn. The graphics were simple wireframe rooms and objects and the user needed to be strapped to the device. It is also considered a precursor of augmented reality, since its transparent lenses let you catch a glimpse of the surrounding environment.

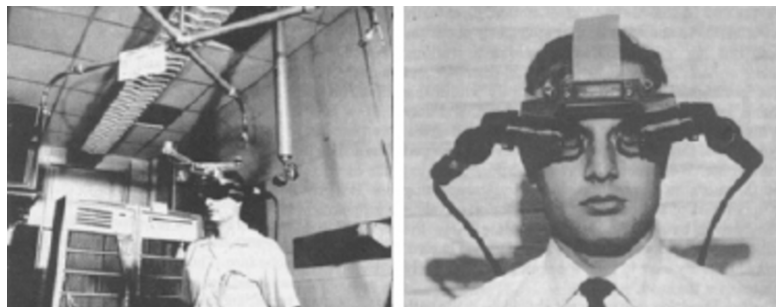


Figure 2.3: The Sword of Damocles  
(url: <https://rb.gy/qkpw7y>)

In the following years, the VR industry mainly focused on devices for medical, flight simulation, automobile industry design and military training purposes. In 1977 the MIT (*Massachusetts Institute of Technology*) created the *Aspen Movie Map*: this system allowed people to do a virtual tour of the city of Aspen (Colorado), in three different modes (summer, winter and polygons). The first two versions used videos recorded from a moving car to give the impression of wandering through the city; the third option was a digital polygonal reconstruction of the city. This device could be defined as the precursor of the modern *Google Street View*.

The term “virtual reality”, however, was coined and diffused only in the late 80's, when Jaron Lanier founded the *VPL Research (Virtual Programming Languages)*, giving a name to this branch of research. This company produced a range of equipment and accessories and in 1984 it was the first company to sell goggles and gloves for virtual reality. In 1989, NASA created the project *VIEW (Virtual Interface Environment Workstation)*, a system for astronauts training consisting in an HMD and gloves for an accurate simulation of tactile interaction. The technology of these gloves led, then, to the creation of Nintendo's *Power Glove*.

In 1991 the *Virtuality Group* made a series of arcade games and devices, which consisted of a set of VR goggles, speaker, joystick and steering wheel, meant for amusement arcades. These systems allowed the tracking of the head's movement and of the controller's position through sensors and magnetic fields, replicating the motions in the virtual environment. An example is shown in Fig. 2.4.

In 1992 the *Electronic Visualization Laboratory* built the first cubic immersive room, *CAVE (Cave Automatic Virtual Environment)*, shown in Fig. 2.4: the system consists in four videos projected on the three walls and on the floor, via mechanisms made up by high performance projectors and mirrors. The user receives spectacles equipped with movements sensors: walking loosely throughout the room, the correct perspective is applied in real-time, ensuring a total immersion of the user.

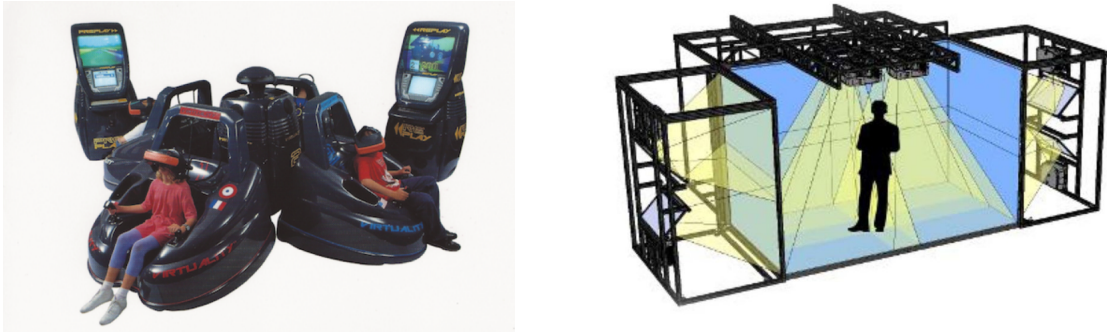


Figure 2.4: a) Virtuality arcade machines ([url: https://rb.gy/1n3ftl](https://rb.gy/1n3ftl))  
b) An example of a CAVE room([url: https://rb.gy/3sjcx6](https://rb.gy/3sjcx6))

The videogame production company Sega Corporation, in 1993, announced the release of *Sega VR*, a VR headset for the console *Sega Genesis*, shown in Fig. 2.5. The prototype consisted in a headset with LCD displays, head tracking and stereo sound. This device never got over the phase of prototyping due to severe problems related to motion sickness and due to potential negative consequences on user's sight, although they had already developed four games for this product. Two years later, Nintendo released the *Virtual Boy*, a 3D gaming console introduced as the first portable console with 3D graphics, shown in Fig. 2.5. The device never reached the hoped sale level and it turned out to be a failure due to the monochromatic displays (black and red), the lack of software support and the poor ergonomics of the product.

The flop of Nintendo's product led the companies to doubt about immersive virtual reality and it took almost two decades of research on the technology for it to gain back the attention of the market. In 2012, on the crowdfunding platform Kickstarter, it was launched a prototype of an headset, which, given the ergonomics and the technical specifications, was destined to change the industry: the *Oculus Rift*, shown in Fig. 2.6. In 2014, the idea was bought by Mark Zuckerberg (Facebook) and, then, put on the market in 2016. In 2014, Google, instead, announced *Google Cardboard*, a paper accessory (therefore, with a reasonable price) able to transform a smartphone in an economic headset, shown in Fig. 2.6. Finally, in the same year Sony announced to be working on a VR headset for the PlayStation 4 (at the beginning *Project Morpheus*, then renamed *PlayStation VR*), available from 2016. Thanks to their potential for the



Figure 2.5: a) Sega VR ([url: https://rb.gy/sltesc](https://rb.gy/sltesc))  
b) Virtual Boy ([url: https://rb.gy/vllwz1](https://rb.gy/vllwz1))

gaming and entertainment industry, these devices were the foundation from which the devices, that are now more popular, were developed.



Figure 2.6: a) Oculus Rift ([url: https://rb.gy/ivn8gg](https://rb.gy/ivn8gg))  
b) Google Cardboard ([url: https://rb.gy/8cbrgu](https://rb.gy/8cbrgu))





Figure 2.7: PlayStation VR  
(url: <https://rb.gy/65ziuv>)

Nowadays, there are two different types of headsets [10], as illustrated below.

- *Tethered VR headsets*: in these systems, the headset is physically connected to a PC through cables, which can be HDMI and/or USB. Due to the higher computing power provided by the PC (high-end GPU and CPU), this technology is the one that deliver a higher quality experience thanks to a better graphic and a faster processing speed. Nevertheless, the wiring affect the users' freedom of movement. Some examples of these headsets are:
  - o *Oculus Rift*
  - o *HTC Vive* and *HTC Vive PRO*
  - o *Valve Index*
  - o *PlayStation VR*
  
- *Standalone VR headsets*:also known as all-in-one HMDs, these headsets are equipped with custom CPUs, sensors, batteries and storage and, therefore, they do not require to be physically linked to a PC to work. The lack of wiring, hence, allow a greater freedom of movements. However, the lower computing power affects the quality of images and refresh rate, while the limited battery influences the usage time of the headset. This type could also be used as tethered, connecting them to a PC via USB cable. Some examples of these technology are:
  - o *Oculus Quest* and *Oculus Quest 2*

- o *HTC Vive Focus*

As it can be seen from this brief overview, each technology has its characteristics and limits: one can determine which system is better depending on the purpose.

## 2.1.2 Input devices

Overall, input devices determine the way users navigate and interact with VR environments and this influences their experience and affects immersion, interaction and presence. In particular, they impact the actions a user can play within the simulation and the ease of use of the system.

Typically, the Human-Computer Interaction (HCI) Loop in VR consists in input devices, which collect the useful data, a computer-aided software, which analyzes the received information and generate a proper feedback, and an output system to deliver these feedbacks to the user.

The basic element of an input system is the tracker, which allow to monitor the position of the user and of the needed physical objects. There is a wide range of tracking devices, that could be divided according to the two properties that follows.

- The way the user enters the instructions:
  - o *Active trackers*:the user operates the commands and the events.
  - o *Passive trackers*:the system operates the commands and the events, tracking user's behaviors and movements.
- The way the information is collected:
  - o In the first case the data provided are absolute: the position and rotation's values are absolute.
  - o In the second case the data are relative: the device tracks the change of information compared to the last measurement.

The aspects that allow to classify these systems and evaluate their efficiency are:

- *Range*: volume within which the tracker can measure position and rotation accurately.

- *Resolution*: minimum difference of position and rotation that the tracker can detect.
- *Accuracy*: error that the tracker can exhibit (absolute value).
- *Latency*: time in ms between the physical action of the user and the beginning of data's transmission.
- *Update rate*: number of measurements per second that the tracker can conduct (Hz).
- *Size and weight*: the less cumbersome and heavy the device is, the better.

No existing device provides the optimal conditions for every parameter indicated above. Below there is a brief description of some of the tracking technologies that are more common in today's VR systems.

### 2.1.3 Output devices

The second key issue of immersive VR is about the physical perception that the user has of the virtual environment: the human being needs specific feedback. Specifically, technologies are still little advanced regarding taste and smell, while there are particular output devices for sight, such as graphic displays, for touch (force and touch feedback) and for hearing.

## 2.2 Social VR platforms

Social VR is, by definition, an experience in virtual reality focused on socializing with other people. These experiences are something in between real-life gatherings and chat rooms, offering a chance to meet or interact with people virtually, walk around or play with them and even create one's own world from the comfort of your home. In order to use these platforms, all what is needed is to download the specific app. Many platforms can also be used on a PC, it is not always necessary, thus, to have a headset. Nonetheless, with these devices the experience becomes more immersive and realistic for the user.

In the majority of Social VR platforms, one starts by creating or choosing an avatar among the ones put at disposal. In some cases, the choice is limited to a human being which can be customized (skin color, clothes, hair, etc.); in others, one can take on the most disparate forms, such as fantastic creatures, robots, etc., with a realistic or cartoonish style: the avatar is what will represent the user on the platform. Once entered the virtual world, the user can create the own room to invite friends in (which can be private or public) or can join in an existing public room and meet new people from all over the world. Some of these rooms are themed and people can play games, dance, listen to music, etc. there. With virtual reality social interactions can take many forms and a person could feel more or less comfortable: sometimes people are there to socialize and meet new people, some other times they just enjoy exploring and creating new worlds. There is not a wrong way to live these experiences [11].

Social VR platforms existing during the development of this thesis are many, hereafter we report a brief description of these ones taken into analysis in this work.

### 2.2.1 VRChat

VRChat is an open world where users can customize their avatars and the environments through the software developing kit for Unity, provided by the platform itself. This flexibility let a lot of freedom to the users, but, at times, it may be confusing and messy due to technical issues in the rooms created by the users and to the extremely diverse style between a room and the other and between an avatar and the other [3].

In 2020, VRChat also introduced a new visual programming language, *Udon*, which uses a node graph system.

### 2.2.2 AltSpaceVR

AltSpaceVR is structured in spaces called *worlds*, which can be found and accessed by menu or via teleporters inside the worlds. Even in this platform users can customize their avatars and worlds, although the avatars can only be built with fixed provided elements (it is not possible to create new ones outside of the platform and upload them), as for the worlds they can be created on the website put at disposal by VRChat. Some worlds, such as *Campfire*, are bigger and they are built and maintained by the official developers: they are places for the users to meet up and interact with each

other. In the main menu there are lists of the “most popular” worlds of the moment and of the events (ongoing or scheduled) which can be accessed.

The main difference from VRChat is the atmosphere perceived in the platform: on VRChat users have more freedom (especially in terms of creativity), but it may seem a bit chaotic.

### 2.2.3 RecRoom

RecRoom is a game center, which offer the possibility to take part in various group games and to socialize with people. The platform consists in a series of built-in multiplayer games, known as *RecRoom Originals*. The users can find their friends in a public hub room, called *Rec Center*, from where they can access the different games or the rooms created by the users themselves. Even in this case it is possible to customize avatars and worlds, although avatars can only be made up with fixed elements (it is not possible to create new ones outside of the platform and upload them), as for the worlds they can be personalized inside the platform with objects provided by RecRoom or built by other users. These objects, known as *Inventions*, are made inside the application with a tool in the menu, the *Marker Pen*.

This platform differs from the previous ones, for it is focused mainly on playing games, instead the others are above all places to socialize.

### 2.2.4 Mozilla Hubs

Mozilla Hubs is a virtual collaboration tool, that, due to its web-based nature, allow to integrate many different platforms into its interface, facilitating the media sharing (contents such as photos, videos, pdf files, links, 3d models, etc.). The main difference from the other platforms taken into analysis is that created rooms are not public and accessible by anyone, but it is necessary to share the room’s link in order to enter it. The customization of the avatars is very limited (as for AltSpaceVR and RecRoom), while regarding the environments there is a lot of freedom (as for VRChat): in fact, one just needs to model the objects in any 3D modeling software (Rhino, Maya, Blender, etc.), export them as *.obj*, and upload them on the *Sketchfab* website. Once they are uploaded, they will be accessible from the platform for the creation of the environments provided by Mozilla Hub, *Spoke*.

This platform was created mainly as a collaboration space; however, it also features a social dimension due to the interpersonal interactions available. Though, it is more difficult to reach and meet new people due to the complexity in sharing rooms.

All of the platforms listed above are accessible from different devices, since they are available also in a *desktop mode*, that let one control the avatar via mouse and keyboard or a gamepad. However, this version has limitations compared to the one for VR headsets, because there is not the possibility to move avatar's limbs and to perform interactions that require more than one hand.

## 2.3 Non-verbal communication in Social VR platforms

In this section we will investigate the use of non-verbal communication in existing Social VR platforms, how it is perceived by users and how it could potentially be improved.

### 2.3.1 Talking without a voice

An analysis [4] conducted in 2020 focused on exploring communication dynamics in digital social spaces, trying to understand the use of non-verbal communication in Social VR platforms. The research is made of two empirical studies: the first collected data to explore the types of non-verbal interactions used naturally in Social VR platforms (similarly to an offline face-to-face setting); the second study, instead, was an interview to investigate people's perception of non-verbal communication in these platforms and how it affects interactions outcomes. As this study highlights, in prior literature just a few explored new practices of NVC in Social VR platforms and their influence on users' interaction. So, the goal of this research is to address limitations in previous literature on NVC dynamics in digital social spaces and to provide evidence of the role and influence of this kind of communication on interactive experiences.

For the first study they collected data on existing platforms, observing and recording users' activities and interactions from the outside, without taking part in them. They, then, analyzed this data and classified recorded non-verbal behaviors, as shown in Fig.2.8. Findings show how NVC are mainly used the same way as in an offline context (such as nodding and applause), but sometimes emojis and actions, such as dancing, were used to highlight people's feelings.

Context	Type of behavior
Indicating attention	Nodding, Head movement/gaze direction, hand gestures
Approval	Emojis, applause
Directing attention	Pointing, patting chest
Social grooming	Waving, dancing, kissing
Interpersonal provocation	Poking, pushing, moving too close
Social disruption	Flying, extensive movement
Entertainment	Dancing, emojis, playing with objects

Figure 2.8: Examples of types of observed non-verbal behaviors [4]

The second study was used to understand how and why users conducted these

behaviors. Results from the interviews' analysis reveal that users perceive NVC as a more immersive and embodied interactions, similar to offline face-to-face ones and a natural way to start communications with strangers. In terms of consequences, findings highlighted that NVC allows privacy, social comfort and protection, since it gives the possibility to avoid using voice.

Based on these results, they, then, outline three potential design implications to further support non-verbal communication and marginalized users in VR:

- supporting facial track;
- more accurate hand and finger tracking;
- alternative modes of control, in order to make NVC on these platforms more accessible, allowing users to control their avatars without physically mirror their bodies in the offline world.

### **2.3.2 Social VR platforms group walk-through**

In 2021, the University College London has carried out a study [3] to examine the usability choices made in existing Social VR platforms, by having a guide to walk participants through a series of social tasks that are common across these platforms. Considering that this is an evaluation on general usability, many aspects have been taken into account, listed below and shown in Fig. 2.9.

- *Identification*: actions to identify a target.
- *Communication*: verbal and non-verbal communication.
- *Navigation*: locomotion in the space and between rooms.
- *Manipulation*: use of interactive objects.
- *Coordination*: gather people and plan actions with them.



**TABLE 2 |** The full set of tasks within each of five main task groups (left column), and which interaction cycle they use for each platform.

	VRChat	RecRoom	AltSpaceVR	BigScreen	Spatial	Mozilla hubs
<b>Identification</b>						
Identify others	Explore	Explore	Goal +3D	Explore	Explore	Explore
Identify speaker	Explore+2D	Explore	Explore+2D	Explore+2D	Explore+2D	Explore
Identify interactor	Explore	3D	Explore	Explore	Explore	Explore
<b>Communication</b>						
Emotion express	2D	2D	2D	-	-	2D
Gesture communicate	Collaboration/2D	Collaboration	Collaboration	Collaboration	Collaboration	Collaboration
Mark friends	2D + Switch	2D + Switch	2D + switch	-	-	-
Text communicate	System+2D	System+2D	2D	-	-	System
<b>Navigation</b>						
Group gathering	Goal	Goal	Goal	Goal	Goal	Goal
Group moving	Goal/2D	2D	Goal/2D	2D	2D	-
Room transport	2D	2D/Switch	2D	2D/Switch	2D/Switch	Switch
<b>Manipulation</b>						
Creating objects	-	2D+3D	-	2D	2D	2D
Moving objects	3D	3D	3D	3D	3D	3D
Passing objects	Collaboration	Collaboration	Collaboration	Collaboration	Collaboration	Collaboration
<b>Coordination</b>						
Room creating	2D	2D	2D	2D	2D	-
Invite others	2D + System	2D/Switch	2D + System	Switch	2D + Switch	Switch
Public room meeting	2D + Goal	2D + Goal	2D + Goal	2D + Goal	-	-
External source sharing	-	-	-	Switch	Switch	-
System notification	System	System	System	-	-	-

Figure 2.9: Set of tasks considered and the interaction cycle they use in each platform [3]

However, for this thesis project the most interesting results were the ones regarding communication. Findings underline how non-verbal communication is significant in Social VR and that it could enhance engagement and improve the user experience. The study also notes how the focus of research in this field is mainly in trying to capture more facial expression and body movement, but that this could bring out an issue of asymmetry between users, who may not have the same capabilities (in terms of tracking and mirroring devices). It, lastly, emerges that, at the time this walk-through was conducted, only Rec Room had a specific interaction for high fives and handshakes.

### 2.3.3 How do I make this thing smile?

A study [6] conducted in 2020 at UC Irvine undertook an inventory of ten Social VR platforms to better understand the design space for non-verbal communication within them and identify opportunities and challenges for future design and research. They divided the inventory in four categories, listed below and shown in Fig. 2.10.

- *Moving and proxemic spacing*: available locomotion modes.
- *Facial control*: facial expressions, both preset and puppeteered, and gaze.



interactions control. Most of the recent studies focus on improving tracking, by introducing new input devices and algorithm. For the thesis it was decided, instead, to focus on alternative modes of interactions control. Below, two examples of case studies that introduce trackers and machine learning to provide more accurate posture tracking are reported.

### **2.4.1 Pose estimation with gloves and trackers**

The first case study [12] is the development of a Social VR platform that supports full-body motion tracking, finger-tracking, motion capture of facial expressions and eye-tracking, thanks to the use of gloves and trackers, to provide a solution for the inclusion of gaze, finger/hand gestures and facial expressions on these platforms. To detect head motion they used a HTC Vive Pro HMD with HTC Vive Base Stations. HTC Vive Trackers are used to track the motion of the user's hands, feet, and pelvis. The motion data captured are integrated into the inverse kinematics solution Final IK, to calculate the posture and transfer it to the avatar. The user's finger motion is captured using Manus VR gloves. For the integration of direction of gaze into the system, they used an eye-tracking setup, compatible with the HTC Vive Pro HMD as an add-on. Finally, to integrate facial cues into the system they used a small infrared camera placed in front of the HMD, which captures facial movements of the lower face of a user. The Social VR system architecture is shown in Fig. 2.11.

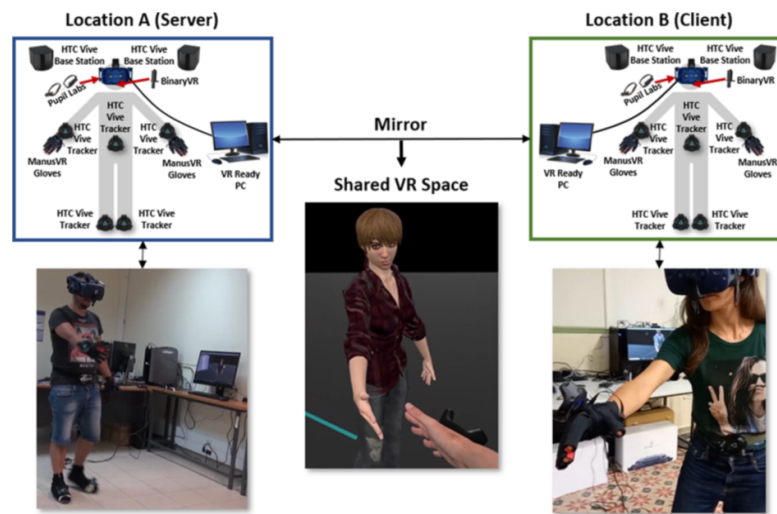


Figure 2.11: SVR system architecture incorporating non-verbal cues [12]

## 2.4.2 Pose estimation with optical camera and machine learning

The second example [13] is a study to detect hands and body gestures using human pose estimation based on off-the-shelf optical camera images using machine learning. The system starts with a webcam image, process it with a gesture tracking algorithm using OpenPose and detect some human key points, as shown in Fig. 2.12. Then, with a neural network model, the pose acquired is compared with a selection of gestures included in the study and is classified. A 3D avatar will, lastly, mirror the detected gesture.



Figure 2.12: OpenPose human pose detection with image (left), without image (right) [13]

### 2.4.3 NVC classification and their presence in Social VR existing platforms

Starting from previous analysis of non-verbal communication in social VR [4, 3, 6], three categories of non-verbal expressions has been defined:

- *Facial control*: it comprehends every facial movement or expression that reveals information. We included in this class *expression presets* (such as emoji or automatic reactions of the avatar), *puppeteered expressions* (facial movements controlled by the user as lip sync) and *gaze* (eyes movement).
- *Gesture and posture*: it incorporates every other movement of the upper body (conscious or unconscious). This category is subdivided in *movable appendages* (control over arms and hands), *mood presets* (mainly emotes that involves avatar's movement) and *movement depending on avatar status* (movements and poses non directly selected or made by the user).
- *Other NVC characteristics*: other movements that allow the user to express feelings without talking. We placed in this class *personal space consent* (possibility to keep people away), *collisions with other avatars* and *legs control*.

The Social VR platforms described previously have been, then, tried out to understand which types of non-verbal communication they feature and how they include them, since the prior walk-through considered were done a bit less than a year before and they could partially be outdated. Below is a table (Fig. 2.13) to summarize this perusal.

State of the art

	VRChat	AltspaceVR	Rec Room	Mozilla Hubs
<b>Facial control</b>				
Expression Preset	Emoji, no facial expression	Emoji, no facial expression	Emoji and reactions (with avatar movements)	Emoji, no facial expression
Puppeteered Expressions	Mouth does not move	Lip Sync	Lip Sync (stylized)	Mouth does not move
Gaze (Eyes movement)	No independent eyes movement	Sight independent eyes movement (no eyes tracking)	No independent eyes movement	No independent eyes movement
<b>Gesture and Posture</b>				
Moveable Appendages	Arm and Hands position (not direct control on fingers)	Arm and Hands position (not direct control on fingers)	Arm and Hands position (not direct control on fingers) - Handshake (with trigger) and high-five	Arm and Hands position (not direct control on fingers)
Mood Preset (Emotes)	-	-	A vatar moves depending on Emoji and reaction selected	-
Movement depending on avatar status (indirect selection)	Walking and poses depending on avatars Fingers position depending on button pressed	Fingers position depending on button pressed	Fingers position depending on button pressed	Fingers position depending on button pressed
<b>Other NVC characteristics</b>				
Personal space consent	Not existing option	You can activate your bubble to keep people at a certain distance	Not existing option	Not existing option
Collisions with other avatars	Collisions with avatar not handled	Collisions with avatar not handled	A vatar collision as objects collision: can not touch other avatars	Collisions with avatar not handled
Legs control	Legs pose estimation	No legs	No legs	No legs

Figure 2.13: Social VR platforms analysis (March 2022)

# Chapter 3

## Technologies

The first step in the design of this project is to choose the most suitable hardware and software solutions for its implementation. In this chapter the used technologies will be briefly introduced.

### 3.1 Hardware

Regarding hardware technologies, the image below (Fig. 3.1) shows a comparison of the characteristics of the main headsets on the market. Given the goal of integrating the most effective proposals in the existing Social VR platforms, the most significant characteristics of the headsets are the cost and the portability of the device, because it has to be a technology easily accessible for the public. Based on these elements, the choice fell on the Oculus Quest 2, the most lightweight and economic device between the proposed ones: thanks to the possibility to use it in standalone mode, it can be used without high-performance PC and in many different contexts.

					
	Oculus Quest 2	Oculus Rift S	HTC Vive Pro	HTC Vive Cosmos	Valve Index
TYPE	Standalone VR	PC VR	PC VR	PC VR	PC VR
DISPLAY	Single Fast Switch LCD (binocular)	Single LCD (binocular)	2x AMOLED	2x LCD (binocular)	2x LCD (binocular)
REFRESH RATE	120 Hz	80 Hz	90 Hz	90 Hz	144 Hz
FOV (horizontal)	97°	88°	98°	97°	108°
WEIGHT	503 g	500 g	550 g	702 g	809 g
BATTERY LIFE	3 hours	PC-powered	PC-powered	PC-powered	PC-powered
DEGREE OF FREEDOM	6 DoF inside-out via 4 integrated cameras	6 DoF inside-out via 5 integrated cameras	6 DoF marker-based	6 DoF inside-out via 6 integrated cameras	6 DoF marker-based
PRICE	399\$ (with controllers)	399\$ (with controllers)	599\$ (only headset)	699\$ (with controllers)	749\$ (with controllers)

Figure 3.1: Headsets comparison  
(data from url: <https://rb.gy/kxvby>)

Another feature that has oriented the choice towards this headset is the integration of handtracking without controllers provided by *META*, which, in the case of future developments, would allow the implementation of a handshake without the need of controllers.

### 3.1.1 Oculus Quest 2

The Oculus Quest 2 (now Meta Quest 2) is a virtual reality headset developed by *Meta Platforms*, released on the market in October 2020 and shown in Fig. 3.2. It is a device that can run as either a standalone headset, with a battery that ensures 2 to 3 hours of use, or with Oculus-Rift compatible VR software running on a desktop computer, connected through a USB-C cable. It is, also, possible to connect it to a pc to stream PC VR contents via the provided wireless connection, *Air Link*. This headset is an updated version of the previous Oculus Quest: the design is similar, but the outer shell is entirely made by white-colored plastic. Additionally, it is lighter in weight than the first-generation Ques (503 g, compared to the 571 g of the former one) and improved in terms of refresh rate, per-eye resolution and controllers.





Figure 3.2: Oculus Quest 2  
(url: <https://rb.gy/9weoj8>)

It features an *Android-based* operating system, with a *Snapdragon XR2* SOC by Qualcomm with 6 GB of RAM. Inside, instead, it comes with a single fast-switch LCD panel with a resolution of 1832x1920 per-eye and a refresh rate up to 120 Hz. The lenses are Fresnel type with an adjustable physical interpupillary distance (IPD), that can be regulated by physically moving the lenses in three possible positions (58 mm, 63 mm and 68 mm). The field of view (FOV) is around 90° both horizontally and vertically. It, then, have two speakers placed in the side arms of the headset, but you can also connect earphones through a lateral jack input.

The movement tracking system has 6 degrees of freedom (DoF) either for the head and for the controllers, thanks to the 4 cameras located on the shell of the headset (inside-out) and to other sensors present on the inside. It can be used in stationary mode or in motion, defining at power on an area of play, outlined by virtual walls that are shown when the user gets closed to them to warn about potential obstacles.

As regards the controllers, the ones included in the Quest 2 are third-generation Oculus Touch controllers, which feature a few different buttons, a thumbstick and partial finger tracking. The controllers' details are shown in Fig. 3.3. However, you can interact with the headset also without controllers, through the featured hand-tracking technology: due to the external cameras, the headsets recognize hands' movements and some specific gestures allow to perform a few actions, such as selecting an item, accessing the various menus and scrolling them.

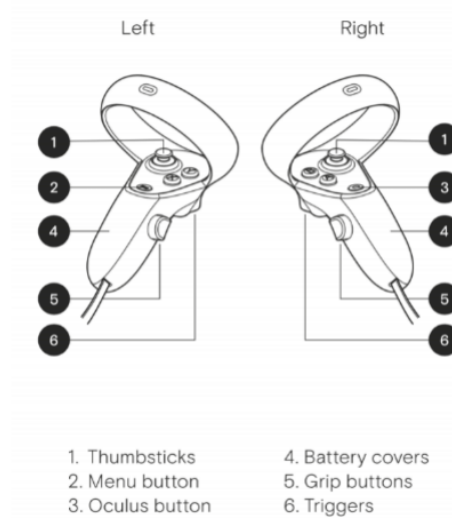


Figure 3.3: Oculus Quest 2 controllers  
(url: <https://rb.gy/ddzsjs>)

## 3.2 Software

In this chapter the software that have been used to develop the thesis' project will be presented: Unity 3D (version 2020.3.41f1) with the XR Interaction Toolkit plug-in and the Photon Pun 2 package, Visual Studio and Blender.

### 3.2.1 Unity 3D

Unity is a cross-platform game engine for the development of 2D and 3D interactive contents and videogames, shown in Fig. 3.4. It supports a variety of desktop, console and mobile platforms, both for VR and AR. This software allows to build a virtual environment integrating sound and visual effects, physics systems, 3D animations and models (made with an external modelling software, such as Maya, Blender, etc., exporting the files in *.fbx* or *.obj* format). It has been adopted also outside of the video game industry, by companies in architecture, engineering, film, construction, etc. The development engine has been implemented in C/C++ and it supports code written in C and Javascript. It consists of two subsets: the *Unity Editor*, which regards the GUI and the interactive aspect, and the *Unity Engine*, which handles the physics, the rendering and the scripting mechanism.

Unity is available either in a Free version and in a Pro one, which has access to more technical support and more customization. It also features an *Asset Store* where the creators can upload and sell or give for free assets or plug-in. An example is the Photon Pun 2 package introduced below.

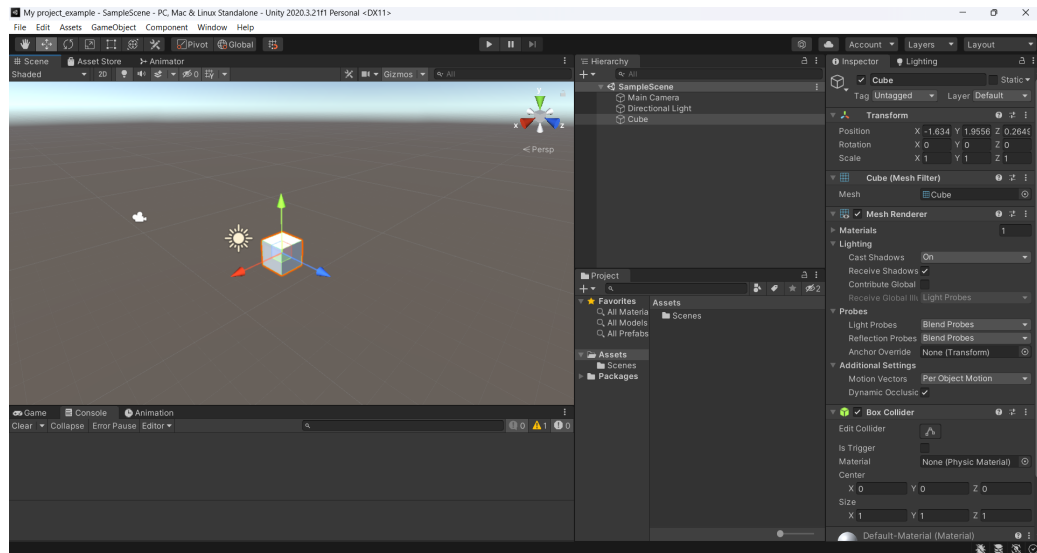


Figure 3.4: Unity interface

### 3.2.2 XR Interaction Toolkit Unity plug-in

The XR Interaction Toolkit [14] is a high-level interaction system, based on components. It lets you control and manage the 3D and UI interactions of the controllers from Unity inputs events. The base structure of this system consists in *Interactor* and *Interactable* components and in an *Interaction Manager* that connects these two types of components. There are, then, additional helper components that can be used to extend these functionalities and include your own interaction events.

These components of the XR Interaction Toolkit supports:

- Cross-platform XR controller input
- Basic object hover, select and grab
- Haptic feedback through XR controllers

- Visual feedback to indicate a possible or active interaction
- Basic canvas UI interaction with XR controllers
- A VR camera rig to handle stationary and room-scale VR experiences

### 3.2.3 Photon Pun 2 Unity package

*Photon Unity Networking* (PUN) [15] is a package for Unity available on the Asset Store for multiplayer games. Photon makes servers available to its users, offering a fast and reliable communication, so that clients do not need to connect one to one. This package holds three level of API's:

- *The PUN code*: it is the higher level, which implements features specifically for Unity.
- *The Realtime API*:it includes the logic to use the servers, do matchmaking, call-backs, etc. It is the intermediate level.
- *DLL files*:they are the lower lever and contain serialization and deserialization, protocols, etc.

Photon provides a flexible matchmaking, that arranges the players in rooms where the objects can be synchronized over the network. Photon offers, moreover, other types of packets, such as Voice and Chat, that allow to add features to the multiplayer game like voice interaction between users. This package is available for free, but it also has a Plus option, that includes more storage on the servers. For this project it has been used the latest release of the free version (Photon PUN 2).

### 3.2.4 Visual Studio

Visual Studio is an *integrated development environment* (IDE) implemented by Microsoft. It is used to develop computer programs, websites, mobile and web applications. It supports different programming languages, including C++, Javascript and Java: in this project we adopted C#. The environment includes a code editor supporting IntelliSense, that provides suggestions to code completion and it reports syntax and logic errors. Among the many features of Visual Studio, it is present also a debugging

option, which allows to check that the game behave consistently with what is written in the code. This IDE has been chosen to write the code of this project because it is the one suggested by Unity and, therefore, it permits to better implement the libraries from Unity itself, providing a proper support.

### 3.2.5 Blender

Blender is a free and open-source software for the creation of 3D contents, maintained by the Blender Foundation. It supports the entire 3D development pipeline: modeling, uv mapping, shading, rigging, animation, physics simulation, rendering, compositing, motion tracking and even video editing. Fig. 3.5 shows the software interface. Furthermore, you can customize the application by the API for Python scripting, that allows also to create new tools. The most used and consolidated scripts among the users are, then, implemented and included in the software's future releases. Blender allows to import and export 3D models created in other software in different formats. Regarding the Render Engine, instead, there are three types available, with different characteristics:

- *Workbench* is the simplest render engine, used for previews during modeling and animation. It does not provide photorealistic results, but it is immediate.
- *Eevee* is a real-time engine, implemented in OpenGL and integrated starting from the version 2.8 of the software. It is not a ray-trace engine: instead of computing every ray of light, it uses rasterization, that does always provide physically accurate results. However, it delivers good renders in a short amount of time.
- *Cycles* is a ray-trace render engine, that allows to achieve photorealistic and accurate renders in terms of physics. Although, it requires longer rendering times.

Blender, in addition, allows the use of external render engines. As mentioned above, this platform supports also rigging and animation of 3D models, through the usage of skeletons, bones and armatures, animated by key frames and interpolated by the software. You can use both direct and inverse kinematics. Lastly, Blender gives the possibility to run physics simulations of fluids, smoke, particle systems, clothes and soft bodies.

The contribution of the community is essential for the update of the platform and for bug fixing, since it is a community-driven project under the GNU General Public License (GPL). Blender is cross-platform and runs well either on Linux, Windows or Mac computers. Its interface uses OpenGL. In this project it has been used for modeling the virtual world and the NPCs (non-playable characters), exported in .fbx format to be, then, imported in Unity and Substance Painter.

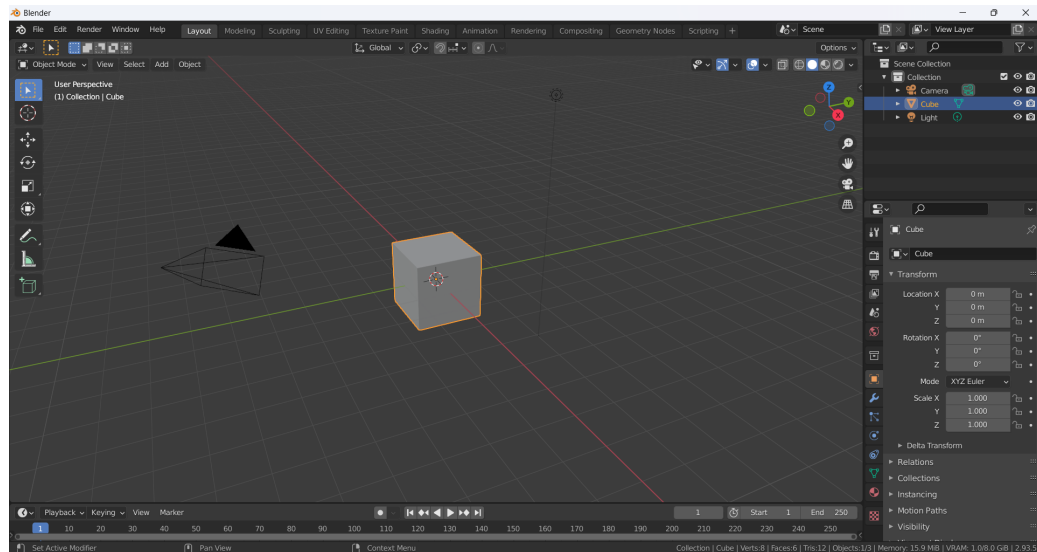


Figure 3.5: Blender interface

### 3.2.6 Substance Painter

Substance Painter, part of the Substance Suite developed by Allegorithmic and then acquired by Adobe, is a 3D painting tool, whose main purpose is to texture models. Its masking and procedural texturing tools permits to obtain textures that could hardly be reproduced in 2D programs as Photoshop. It allows to paint either on 2D maps and directly onto the 3D object in the viewport. This means that you can see the edits in real-time so you know exactly how the texture will turn out on the final model. This software, in fact, shows and exports all the textures in a PBR format in order that, once imported in the game engine, they will have the same result shown in the painting software. Before importing models into Substance Painter, though, they have to be properly UV mapped: the 3D assets are, then, used as the surface to paint.

This platform is available with a license fee, however there is also a free license for students. The interface is shown in Fig. 3.6.

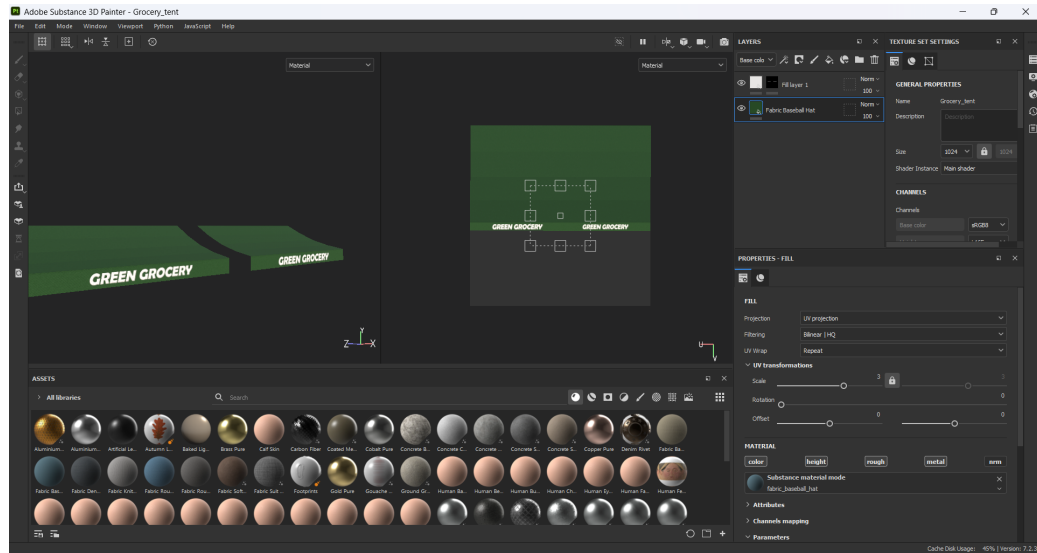


Figure 3.6: Substance Painter interface

# Chapter 4

## Design

In this chapter the features and functionalities of the implemented system will be shown, starting from the decision that led to the chosen design.

### 4.1 Design

The literature review highlighted how previous studies on non-verbal communication in Social VR platforms were mainly of two types: a walkthrough and comparison of the existing platforms and of the NVC opportunities available within these platforms [4, 3, 6]; or else a proposal of new ways to non-verbally interact that involved the development of additional tracking and feedback devices [16, 13, 12].

For this thesis project, instead, the purpose was to explore and propose new ways to non-verbally communicate that exploited only the controllers provided with the headsets, so that they could be integrated in existing platforms and used by the public without the need to purchase, build or install new components.

In an initial phase of the design, three typologies of interaction has been taken into consideration: handshake, high-five and hug, since they are the most common in a stage of meeting and networking between people and they are integrated just in a few platforms. Later, some key points to be taken into account during the design of the proposals and during their implementation have been defined:

- The only tools at disposal for the users to perform the actions are the controllers



- Not having the availability of *force feedback*, it is necessary to take full advantage of haptic, sound and visual feedback provided.
- For all the three actions, keeping a first-person view could be more engaging for the user; at the same time, it could be difficult to understand what is happening during a hug in a first-person view.
- Users' consent must be taken into consideration: someone may not be willing to take part in the interaction. Hence, it is necessary to include a request for consent or to provide to the users the possibility to enable/disable non-verbal interactions.

Then, were defined the parameters that can be monitored and, thus, use for the implementation:

- The *distance* between avatars.
- The *position and rotation of the hands* in space (absolute and with respect to the other user's hands).
- The *position and rotation of the head* in space (absolute and with respect to the other user's hands).
- The pressure on controllers' *buttons*.
- The *hands' movement*.
- The controllers' *vibration*.

#### **4.1.1 Handshake Options**

Three interaction options have been outlined for each action considered, for a total of nine gestures. However, only one of the interactions was explored, developed and tested, the handshake, because the mechanics were similar between the three different actions. To have a wider range of options to compare, it has been added, then, a fourth option of handshake. Based on the results obtained by tests, the implementation can then be extended also to the high-five and the hug.

The four typologies that were defined are described in the following.

- *Handshake 1*: for this first type, it has been used a mechanic that is already present in Social VR platforms and used for emojis. In every examined platform, in fact, one can choose from a specific menu which emoji to use and, once he selects it, an animation is triggered, that either shows the image to the other users or make to avatar do a special movement. An example is provided in Fig. 4.1.

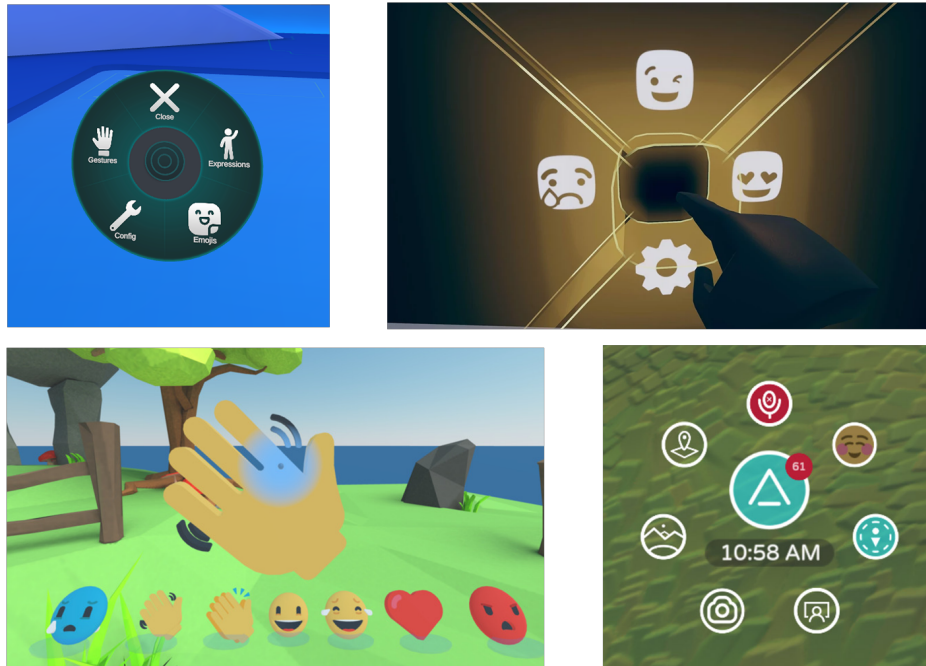


Figure 4.1: In order: VRChat Action Menu, RecRoom Espresso Menu, Mozilla Hubs Emoji Spawners, AltSpaceVR Radial Menu

In this option, therefore, when one approaches another user, a menu shows up on your active hand, which invites him to press the lower button on the corresponding controller to activate the handshake. Once he presses the button, a pop-up message will appear to the other user, asking for consent to handshake. If the user will accept, an animation involving the right hand of both people will be automatically triggered. This option could be defined as non-simultaneous and automatic.

- *Handshake 2*: for the second kind, the aim was to modify the first option, making

it immediate and intuitive for the user, without straying too far from it. In this case, in fact, in close proximity to another user a message appears, inviting to press the button A on the right controller to activate the handshake and to keep it pressed until the other user press it as well. Then, the other user will see a request to press the button A to return the handshake. Only when both will be pressing the button at the same time, an animation involving the right hand of both people will be automatically triggered. This option could be defined as simultaneous and automatic.

- *Handshake 3*: the third kind, instead, is a hybrid between the second and the fourth. In this case, when one approaches another user, a message invites him to grab the other user's right hand, by pressing the grab button, to perform the handshake. Once he grabs it, the movements of his right hand will be frozen and a text will ask him to wait for the other user to grab his hand. On the other side, the other user will be informed that he wants to handshake that user and asked to grab his hand to return the interaction. Once the second user grabs the first user's hand as well, an animation involving the right hand of both people will be automatically triggered. This option could be defined as simultaneous and both non-automatic (since to start the handshake you have to physically grab the other user's hand) and automatic (because of the animation that is triggered by the interactions).
- *Handshake 4*: for the fourth type, lastly, it has been found a more immersive solution for the user, avoiding automatic animations. When a user approaches another user, a message invites him to grab the other user's right hand, by pressing the grab button, to perform the handshake. Once he grabs it, the movements of his right hand will be frozen and a text will ask him to wait for the other user to grab his hand. On the other side, the other user will be informed that the first user wants to handshake him and asked to grab the first user's hand to return the interaction. Once he grabs his hand as well, they will be able to freely handshake, physically moving the virtual hand of the other user. This option could be defined as simultaneous and non-automatic (handshake directly controlled by the user).

Some features are common between the four proposed types of handshakes, so that the transition from one to the other is not alienating for the user and it is easier to

switch between them once he/she is familiar with the interactions.

#### 4.1.2 Test scenario

It was, then, necessary to design and define a scenario appropriate for the tests, to allow users to use handshake in a context as natural as possible and similar to the one of Social VR platforms. Hence, the environment was designed as a town square, where there are two NPCs representing the mayor and a waitress. The user is welcome by the mayor, who introduces him as a new residence of the village: the user can, then, test the handshake many times introducing himself to the shopkeeper and his fellow citizens. In this situation the other citizens are the other users connected at that time. The NPCs are automatically activated when the user gets closed to them, they ask the user to handshake and talk to him/her. The mayor, after his speech, goes away allowing the user to handshake him only ones. With the waitress, instead, you could shake her hand every time you get close to her, since she offers you her hand each time. The image below (Fig. 4.2) is a sketch of the square floor plan and the location of the NPCs within it. These are distinguished according to the typology they belong to: the dot is the waitress, triggered by distance; the star is the mayor which is automatically awoken when the user is spawned.

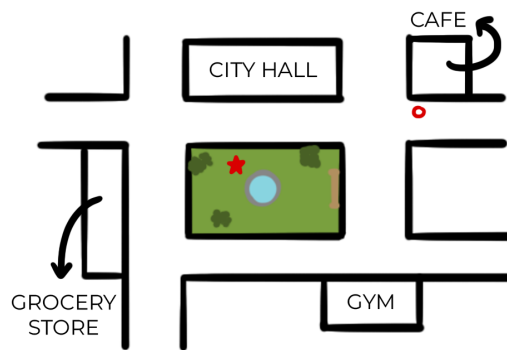


Figure 4.2: Sketch of the environment for the test scenario

Then, users' behavior was monitored during the test and they have been administered some questionnaires to evaluate their experience and their preference between the proposed options of interaction. The methods of measurement and the results will be discussed more in detail in Chapter 6.

# Chapter 5

## Development

In following section, the development of the project and its preparation for the user study will be presented step by step. It starts with the setup of the base VR scene, next the setup network for the multiplayer system, then we will describe the implementation of each handshake and, lastly, we will explain the construction of the test scenario: world building, tutorial rooms and NPCs.

### 5.1 Base VR scene setup

The first phase of the implementation consisted in setting up the base VR scene. The Unity plug-in XR Interaction Toolkit, described in Chapter 3, provides a series of classes and components that implement base mechanisms to track the movement of headset and controllers and to handle the interaction with scene objects. The main component that controls the tracking is called *XR Origin*: this has been customized with scripts to add the fingers' animation when the user presses the grip and trigger button. The *Tracking Origin Mode* is set to *Device* and the *Camera Y Offset* is set to a fixed value, so that all users will have the same height when they enter the environment.

### 5.2 Multiplayer setup

The second thing that has been created and configured is the multiplayer system. To simplify its building and management, we used the *Photon Pun plug-in* for Unity, that

provides server for the users to connect to and predefined components in the game engine to keep track of default parameters such as position, rotation and animations of the objects. It is also possible to monitor specific values via script, using observable interfaces, and to create customize RPC (Remote Procedure Calls) methods to enable remote calling for these methods.

In this project it has been integrated also the Voice package provided by Photon to allow connected users to talk as well, in order to facilitate the interaction and to make it more complete.

### 5.2.1 Server connection

To link the Unity project with the Photon servers, it is necessary to create the application on your profile on the Photon Cloud website, changing the settings as needed. In this case, we left the standard settings provided for the free version of Photon Pun, editing the *Regions Whitelist* in “eu;”, in order to ensure that users are placed in the same region and, therefore, connected to the same server. Once one confirms the desired settings, he needs to copy the application id from the website and insert it in the corresponding field in the *Photon Server Settings* panel on Unity. From this window he can set the parameters used from the server connection, such as the used protocol.

To handle the link with the server, a script, *Network Manager*, has been used, where the necessary methods to monitor the connection status are implemented: *ConnectUsingSettings* and *JoinLobby*. Once the user is connected to the server, he will enter the lobby, where he still will not be able to see or talk to the other players, but here he can choose which room to access. In this project the lobby has also been used to display the tutorials before entering the corresponding room.

### 5.2.2 Joining rooms and spawning players

The management of the rooms access and of the transition from one to the other has also been implemented through a specific script, using the methods offered by Photon, *LoadLevel* and *JoinOrCreateRoom*, that enable to load the desired scene and place the user in the corresponding online room.

Once a user accesses a room, a corresponding net player is created and spawned in the scene, in order for it to be visible for any other connected user.

### 5.2.3 Players setup

Every proposed kind of handshake requires that the object associated with the player has different features: for this reason, a different *prefab* for each scene has been created.

The base structure of the object, though, is the same for every interaction alternative and is made up of the following elements.

- An *Empty* container, called *Network Player Hx* (with x varying according to the handshake mode). Various components are assigned to this *GameObject*, including some provided by Photon for parameters synchronization over the network:
  - o *Photon View*: it is the main element, since it identifies the object across the network with the univocal id linked with it. It configures how the client update remote instances. It allows to “observe” the other components offered by Photon or the scripts that implement the interface *IPunObservable*, to monitor the change of some parameters across the net and update them accordingly.
  - o *Photon Voice View*: use to reproduce voice through the net.
  - o A customized script to synchronize net player’s movements with the ones of the corresponding local XR Origin.
- A *GameObject* for the head, which has various children depending on the handshake variant. Associated with this element, there is a Photon component, *Photon Transform View* to make position and rotation coincide across the network.
- An *Empty* called *Left Hand*, synched with the left controller that has the prefab with the hand mesh as a child.
- An *Empty* called *Right Hand*, synched with the left controller that has the prefab with the hand mesh as a child.

- *Colliders* to enable the interaction when users get close to each other, one that activates the instruction UI (user interface) canvas when the collision starts; one that disables those canvas and any started action when the collision ends.
- Another *Collider*, assigned to an Empty object called *Bubble*, to prevent the users to pass through each other: it is a small box around the user's head set to trigger and when another user's *Bubble* collides with it, both users are slightly moved back.

Every time a user connects to a room, a clone of this prefab associated with that user is created. When a user leaves the room, the object is destroyed.

## 5.3 Handshake: Option 1

In this section it is described the operation method of the first option of handshake, the non-simultaneous and automatic one.

### 5.3.1 Player features

The player prefab used for this variant is made up by the base structure described above, with the addition of UI canvas components, to give instructions to player about how the interaction works, and of a "fake hand", shown only for the animation. The guiding panels are placed on the user's hand and under the user's head, as shown in the Figure 5.1.

### 5.3.2 Handshake mechanism

When a user's head enters the activation collider of another user, the function *OnTriggerEnter* is called. This function shows a panel on the hand of the user, that gives him instruction on how to start a handshake with the other person, and provides haptic and sound feedback to better notify the user of the enabled interaction. In addition, a public boolean parameter is set to true, so that every object could keep track of the collision.



When a user's head exits the deactivation collider of another user, the method *OnTriggerExit* is called. This method hides all the active panels attached to the users and cancel every started interaction between them.

To make the user more comfortable using this variant, it has been given to him the possibility to choose which controller to use to propose or confirm the handshake: he can change it by pressing the trigger button of the controller he wants to set to active. When two users are colliding, if one of them presses the lower button on the active controller (X button for left controller, A button for right controller), a handshake request is sent to the other user. The requesting user will see the instructions change to a message that invites him to wait for the confirmation of the other person. The other will see a panel appear under the head of the requesting user, asking to confirm the handshake, and the right hand of the requesting user will turn blue to highlight his waiting state. Thanks to the interface *IPunObservable*, a boolean value keeps track through the network of the activation and deactivation of the confirm panel and the blue hand for every user.

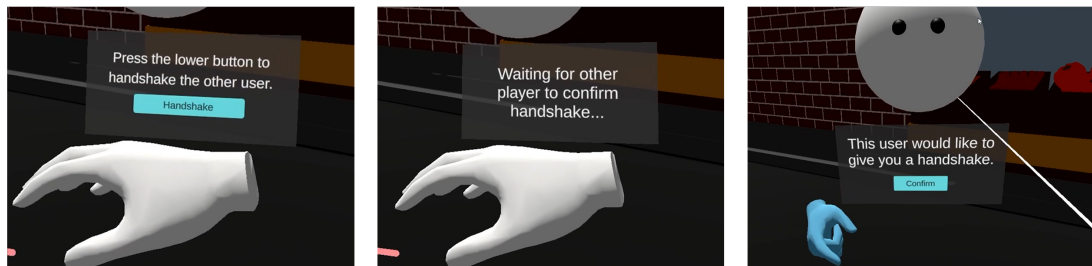


Figure 5.1: a) Starting panel b) Waiting panel c) Confirm panel

If the requested user confirms by pointing at the confirm button with the controller and pressing it with the trigger button, a RPC method, named *ActivateHandshakeOverNetwork*, will be called for every player in the network. This function checks if the player the script belongs to is involved in the handshake, by comparing the ids of the players involved with his. If the player is participating, a handshake animation involving his right hand will be triggered. If a user presses the lower button of the controller while the confirm message under the other's head is shown, the system will record it as an acceptance of the request, so the RPC method will be called and the animation will be triggered anyway.

To undo a sent request or avoid the other user's one, it is necessary to move away from the other person and exit the collision.

The animation is made of three steps. First, using the *DOTween* plug-in's methods, the hands move from the controllers' position to a center position, computed as the middle point of a vector between the users' head on x and z axis, while the y axis is assigned with respect to the lower head. The rotation is set accordingly to the same vector use for x and z axis. Secondly, when both hands reach the middle point, an animation that moves them up and down is triggered. Lastly, at the end of the animation, the hands move back to the controllers' position with the same methods mentioned before. This process is not applied to the users' hands that move accordingly with the controllers: these hands, in fact, are hidden at the start of the animation and "fake hands" are shown and used for the animation, so that there are no conflicts with the controllers' movement.

To monitor the button pression to initiate the handshake, it has been used the default Button Action type of the Unity Input System. While the confirmation was controlled by the `OnClick` function of the button itself.

## 5.4 Handshake: Option 2

In the following paragraphs the structure of the second variant of handshake is explained, the simultaneous and automatic one.

### 5.4.1 Player features

As in the previous variant, the player prefab is made up by the base structure described before, with the addition of UI canvas components, to give instructions to player about how the interaction works, and of a "fake hand", shown only for the animation. In this case the guiding panels are located under the user's head, as show in the Figure 5.2.

### 5.4.2 Handshake mechanism

When a user's head enters the activation collider of another user, the function `OnTriggerEnter` is called. This function shows a panel under the colliding user's head, that gives instruction on how to start a handshake with the other person, and provides

haptic and sound feedback to better notify the user of the enabled interaction. In addition, a public boolean parameter is set to true, so that every object could keep track of the collision.

As before, when a user's head exits the deactivation collider of another user, the method `OnTriggerExit` is called. This method hides all the active panels attached to the users and cancel every started interaction between them.

When two users are colliding, if one of them presses and holds the lower button on the right controller (A button), a handshake request is sent to the other user. The requesting user will see the instructions under the other person's head disappear and a message appears on his right hand, that invites him to wait for the confirmation of the other person. The other will see a panel appear under the head of the requesting user, asking to press the same button (A on the right controller) to confirm the handshake, and, in addition, the right hand of the requesting user will turn blue to highlight his waiting state. In this case, the interface `IPunObservable` is used to keep track of the button pressure for every user in the network, monitoring another boolean.



Figure 5.2: a) Starting panel b) Waiting panel c) Confirm panel

If the requested user confirms by pressing the A button while the requesting user is still holding it, a RPC method, named `ActivateHandshakeOverNetwork`, will be called for every player in the network. This function checks if the player the script belongs to is involved in the handshake, by comparing the ids of the players involved with his. If the player is participating, a handshake animation involving his right hand will be triggered.

If the requesting user releases the button before the other presses it as well, the request is canceled. Another way to undo a sent request or avoid the other user's one is moving away from the other person to exit the collision.

The animation, instead, is made of the same three steps mentioned before.

For this variant, to monitor the A button pression, it has been used the default Button Action type of the Unity *Input System*, adding an Interaction to it with the Trigger behavior set to *Release Only*, so that the system could notify of both the pressure and the release of the button.

## 5.5 Handshake: Option 3

Below it is described the operation method of the third option of handshake, the one that is simultaneous and hybrid between non-automatic and automatic.

### 5.5.1 Player features

The player prefab used for this variant is made up by the base structure described above, with the addition of UI canvas components, to give instructions to player about how the interaction works, and of a “fake hand”, shown only for the animation. The guiding panels are placed on the user’s hand and under the user’s head, as show in the Figure 5.3. It differs from the previous options for a *XR Grab Interactable* component attached to the right hand of the player, that makes it grabbable for the other users. The hand object also has a *Sphere Colliders* that determines the area within it is grabbable.

The *RightHand Contoller* object, moreover, has a *XR Direct Interactor* component instead of a *XR Ray Interactor*. This allows the user to interact with objects that have the *XR Grab Interactable* component attached to them and lets him grab them.

### 5.5.2 Handshake mechanism

When a user’s head enters the activation collider of another user, the function *On-TriggerEnter* is called. This function shows a panel under the colliding user’s head, that gives instruction on how to start a handshake with the other person, and provides haptic and sound feedback to better notify the user of the enabled interaction. A public boolean parameter is set to true, so that every object could keep track of the collision. In addition, the property *allowSelect* of the *XR Direct Interactor* is set to true to enable the interaction with grabbable objects.

Like the previous options, when a user's head exits the deactivation collider of another user, the method *OnTriggerExit* is called. This method hides all the active panels attached to the users and cancel every started interaction between them. It also the property *allowSelect* to false so that the interaction with the other user's hand cannot be erroneously done when the users are far from each other.

When two users are colliding, to initiate the handshake one of them has to put his right hand close to the other user's right hand and press and hold the grab button on the right controller. If one of them does it, a handshake request is sent to the other user. The requesting user will see the instructions under the other person's head disappear, a message appears on his right hand, that invites him to wait for the confirmation of the other person, and his right hand's movement will freeze. The other will see a panel appear under the head of the requesting user, asking to press the same button (grab button on the right controller) to confirm the handshake, and, in addition, the right hand of the requesting user will turn blue to highlight his waiting state. In this case, the interface *IPunObservable* is used to keep track of the button pressure for every user in the network, monitoring a boolean.



Figure 5.3: a) Starting panel b) Waiting panel c) Confirm panel

If the requested user confirms by pressing the grab button while the requesting user is still holding it, a RPC method, named *ActivateHandshakeOverNetwork*, will be called for every player in the network. This function checks if the player the script belongs to is involved in the handshake, by comparing the ids of the players involved with his. If the player is participating, a handshake animation involving his right hand will be triggered.

As for the second variant, if the requesting user releases the button before the other presses it as well, the request is canceled and everything goes back to the previous state.

Another way to undo a sent request or avoid the other user's one is moving away from the other person to exit the collision.

The animation, in this variant, is slightly difference: since users' hands are already closed to the middle distance between them, the animation consists only in the second step. So the users' right hand is hidden, the "fake hands" are shown already in the correct position and the up and down animation is triggered. At the end of the animation, "fake hands" are set inactive again, replaced with the users' "correct" hands.

For this variant, the grab button press is monitored by the *XR Grab Interactable* component that triggers a function every time the hand is grabbed or released.

## 5.6 Handshake: Option 4

In this section the structure of the fourth option of handshake is described, the simultaneous and non-automatic one.

### 5.6.1 Player features

The player prefab used for this variant is made up by the base structure described before, with the addition of UI canvas components, to give instructions to player about how the interaction works. The guiding panels are placed on the user's hand and under the user's head, as show in the Figure 5.4. As the variant number three, it has a *XR Grab Interactable* component attached to the right hand of the player, that makes it grabbable for the other users. The hand object also has a *Sphere Collider* that determines the area within it is grabbable.

The *RightHand Contoller*, moreover, has a *XR Direct Interactor* component instead of a *XR Ray Interactor*. This allows the user to interact with objects that have the *XR Grab Interactable* component attached to them and lets him grab them.

### 5.6.2 Handshake mechanism

This mechanism is very similar to the previous variant, but when both users are grabbing, instead of activating an animation, the right hand's movement are set back to free and users are able to physically shake the other's hand, as shown in Figure 5.4.



Figure 5.4: a) Starting panel b) Confirm panel c) Shaking hands

## 5.7 Test setup

In these next paragraphs, the set up of the scenario for the test is explained, starting from the environment, then introducing the NPCs and, lastly, describing the realization of the tutorial scenes.

### 5.7.1 Virtual environment

The environment where tests are set represents a square of a village, where the user can wander freely. The style for the aesthetic could be defined as cartoonish, low poly and minimal and it has been chosen for the two reasons described below.

- The reference world for this work are Social VR platforms: most of the ones taken into account during the initial analysis have a stylized design.
- The main focus of this study are the proposed handshake systems and their mechanisms, so it was necessary that users were not distracted from the surroundings when testing the interactions.

The main buildings around the square are the town hall and a cafeteria, represented in Fig. 5.5, a grocery store and a gym, shown in Fig. 5.6. These last two structures were designed in case other NPCs needed to be added later. Houses and other buildings have, then, been randomly added to fill up the space. The models and materials were done in with Blender, while the signs were textured with Substance Painter.

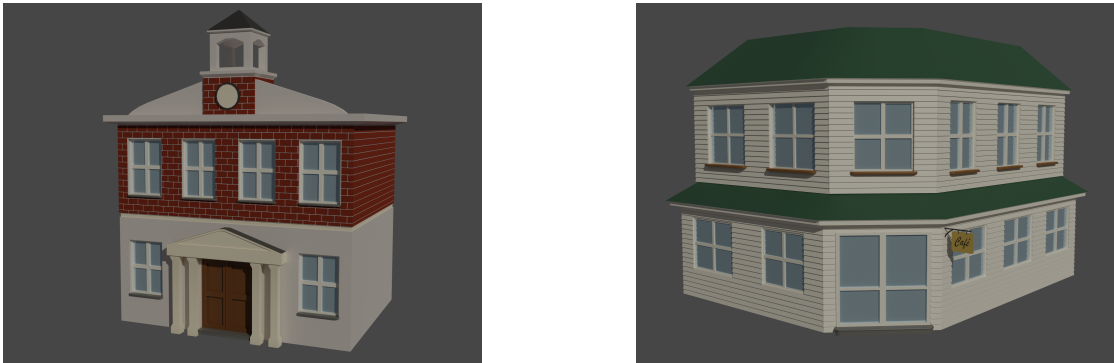


Figure 5.5: a) Town Hall b) Cafeteria

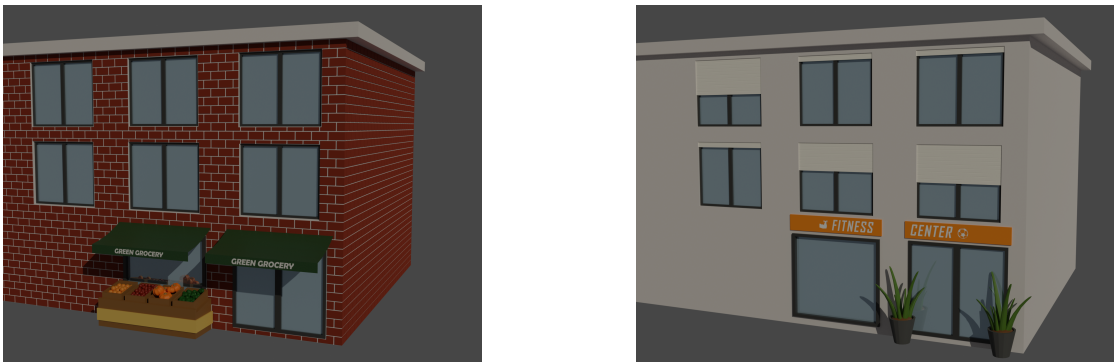


Figure 5.6: a) Grocery store b) Gym

### 5.7.2 NPC

Two NPCs are placed in the environment to help the user get used to the handshaking systems. They are briefly introduced below. Their mechanism description is not examined in every detail, since the handshake system is the same describe above for users.

#### The mayor

This is the character, shown in Fig. 5.7, that welcomes the user at the beginning of each simulation. With a short speech he introduces the player as a new resident of



the town and offers him his hand to handshake. Once the user handshakes him, he tells him to have a walk around and meet the shop owner and the other citizens (other users). When the speech ends, he goes away. This character has been modeled in Blender as well.

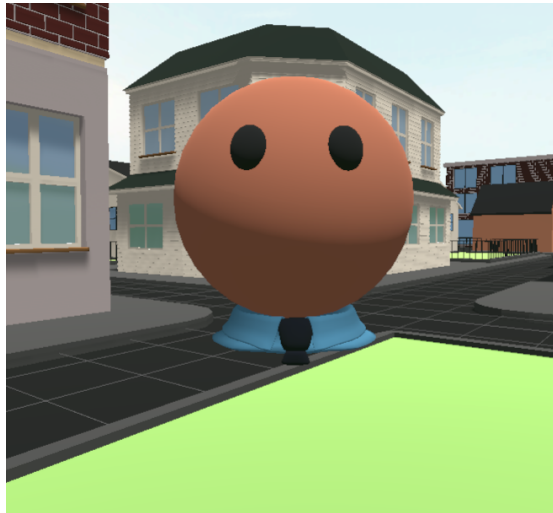


Figure 5.7: Mayor

### **The waitress**

The second NPC, represented in Fig. 5.8, is a waitress placed in front of the cafeteria. When a user gets close to her, entering her collider, she is triggered and offers her right hand to be shaken. At the end of the handshake, an audio clip and an animation are called and she introduces herself to the user and offers him a coffee. Every time the user exits the collision with her, she goes into an idle animation state and cannot be handshaken. As the previous NPC, she has been modeled in Blender as well, while the animations are made directly in Unity.



Figure 5.8: Waitress

### 5.7.3 Tutorial scenes

The first environment that the user enters when starting the test is the lobby, where he will choose which handshake to start with. This place's design recalls a waiting room: it has two sofas, a small table, a lamp and a plant. The buttons that the user has to press to enter the handshake simulation are placed on a wall, as shown in Fig. 5.9. Once the user selects a variant, a video tutorial, that gives instruction on the handshake mechanism, appears on the same wall. When the user feels to have understood how the interaction works, he can start the experience and join the corresponding multiplayer room.

The tutorials consist in videos from a first-person perspective of a user performing the action with another person. These are supplemented with subtitles, arrows pointing at elements to highlight them and a controller image to show the user which buttons to press for the interaction. Each handshake variant has two tutorials displayed before starting the simulation: the first video shows how to initiate the handshake, while the second displays how to respond to the interaction started from the other user.



Figure 5.9: Tutorial scene

# Chapter 6

## User study and results

In the first part of chapter, the experimental protocol and the evaluation criteria used for the tests are described. Then, an analysis of the collected data is reported.

### 6.1 Experimental protocol

The test to evaluate and compare the handshake options was structured as follows.

An initial questionnaire for demographic information was first administered to the testers.

Users were, then, placed in the virtual environment described at Chapter 5, with the goal of introduce themselves to as many avatars as they could. A counter on their left hand displayed how many handshakes they had completed: the aim was to achieve ten successful handshakes, in order to understand how the system works and get confident in using it.

When they reached ten handshakes, they were asked to exit the simulation, remove the headset and answer a questionnaire about the variant experienced. This process has been repeated for all four handshake options for each user. The order of the options during the test has been randomized between users, so that the results were not biased by always using the same order.

When the tester had completed all the four variants, another questionnaire was administered to evaluate the overall experience and compare and rank the four handshake options.

All the questionnaires used are listed in Appendix A.

### **6.1.1 Pre-experience questionnaire**

The questionnaire administered before the test is a demographic inquiry to gain general information about the sample of testers and about their experience with immersive virtual reality and Social VR platforms. The questions are listed below.

- Age
- Gender
- Professional Status
- What is your English level?
- What is your Italian level?
- Do you have experience with programming?
- Have you ever used devices for immersive virtual reality and, if so, how often?
- Do you play videogames and, if so, how often?
- Have you ever used Social VR platforms and, if so, how often?
- Why do you use these Social VR platforms?
- For which kind of activity do you usually use them?
- When you meet a person for the first time, do you usually introduce yourself using the handshake?
- Does handshaking people bother you?
- Does handshaking people make you feel uncomfortable?

## 6.1.2 Post-variant questionnaire

After each variant, a second series of questions are asked to the user, subdivided in five sections: *Embodiment*, *Inconsistencies of Presence*, *Social Presence*, *System Usability Scale* and *Custom*.

The *Embodiment* section is used to evaluate how much the user identifies his body with the body of his virtual avatar. It consists in four statements that the user has to rate from 1 to 7 in terms of agreement (1 as strongly disagree, 7 as strongly agree).

The questions grouped in the *Inconsistence of Presence* are divided in two categories: four of them are asked after each handshake option; the other thirteen are asked just once since they refer to the VR experience more in general and users tended to answer the same thing after each variant. They evaluate the ability of a user to feel that they are actually in a virtual location. They are statements that the user has to rate from 1 to 5 in terms of agreement (1 as strongly disagree, 5 as strongly agree).

The part called *Social Presence*, instead, measure the perception that users have of other persons in the experience as being physical “real” persons. As the previous sections, some of the questions are asked just once, for the same reasons explained before, while eight of them are asked after each handshake option. Again, they are statements that the user has to rate accordingly to his agreement with them, with a scale that goes from 1 to 7 (1 as strongly disagree, 7 as strongly agree).

The section *System Usability Scale* evaluates the usability of the system, that in this case refers to the handshake mechanism. It is made up by 31 statements that the user has to rate from 1 to 5 in terms of agreement (1 as strongly disagree, 5 as strongly agree).

The last part, *Custom*, consists of 9 statements specifically about the handshake that the user has to rate with same scale used before (1 to 5).

## 6.1.3 Post-experience questionnaire

After completing the experience, the user has to fill three other parts of the questionnaire: the *Simulator Sickness Questionnaire* to measure his state of health during the test, a *Quality of Experience* section of five questions to measure the overall enjoyment throughout the simulation and, lastly, a ranking part, where he is asked to rank the four options based on various characteristics, such as the ease to learn, the ease to

use, etc.

## **6.2 Results analysis**

The test was performed on a sample of 15 people and in these paragraphs is a summary of the results obtained.

In order to evaluate the statistical significance of the collected answers, a pairwise signed-ranks test was executed on the data. It calculates a  $p$  value which is usually compared with the standard threshold of 0.05; if the value is lower, the difference between the averages of the two groups is significant, whereas if it exceeds the value then the collected data cannot be used to validate hypotheses.

### **6.2.1 Participants demographics**

The set of participants was made of 12 males and 3 females, with an average age of 25,87 years. All testers are native Italian speakers, but with a good level of English (46,6% with an intermediate level, 53,3% with an advanced level). Only for one participant it was the first time experiencing immersive virtual reality, but just 33,3% of the sample uses it often. Just the 26,6% of the participants had used Social VR platforms before. When they meet a person for the first time, they introduce themselves with a handshake most of the time, with an average score of 3,8 out of 5. The professional status division and other data are shown in Fig. 6.1.

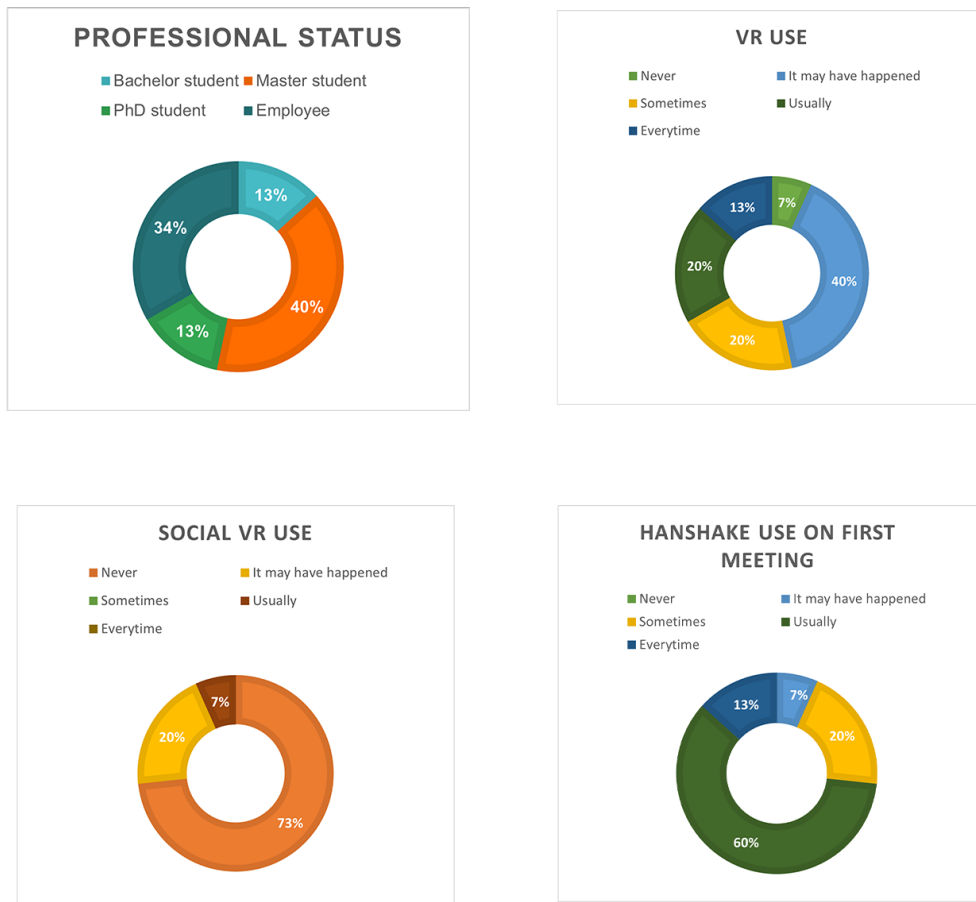


Figure 6.1: Participants demographic data

## 6.2.2 Options comparison

### Overall preference from rankings

The results' analysis shows how the overall preference emerged from the ranking is the option number 4, with a significant difference compared to all of the others in most of the characteristics considered. There are, though, a few exceptions: in the ease in starting a handshake ( $p$  value = 0,123), in the ease in completing a handshake initiated from the other ( $p$  value = 0,231), in the ease in canceling a handshake started from the user ( $p$  value = 0,203) and in the least distracting ( $p$  value = 0,065), there is not a



significance difference between options 3 and 4. This deviation was expected since the only difference between these two variants is the action of shaking, while they use the same mechanism to start and cancel the handshakes. The other peculiarity is shown in the ranking for the ease in canceling a handshake started from the other user, because in this case none of the options present a significant difference from the others, but it is explained by the fact that in all variants to cancel the handshake initiated by another user the mechanism is the same: it is necessary to move away from the other person.

### **Custom section**

In terms of social presence and fidelity, what emerges from the custom questions is that, again, the highest score was obtained by the variant number 4 with a significant difference compared to all the other options. A few users explained these results by adding in the comments that the similarity of this option's movement with movements performed in a real-life handshake made this variant the one that felt more realistic and, therefore, also increased to feeling of interacting with a real person.

### **System Usability Scale**

Regarding ease of use and usability strictly related to the handshake the highest score was obtained by the variant number 4 with a significant difference compared to all the other options. When considering the general usability of the system, instead, there is not a significant difference between variant 3 and 4. Meanwhile, about the learnability the results show that there is not significant difference between the 4th and the 3rd option ( $p$  value = 0,773) and between the 3rd and 2nd option ( $p$  value = 0,089), while the difference is significant between the 2nd and the 4th option ( $p$  value = 0,048), labeling the 4th as easier to learn than the 2nd. The 1st option presents a significant difference from all the other variants with regards to these measurements.

### **Embodiment, Inconsistence of Presence and Social Presence**

The results for the overall embodiment, social presence and inconsistence of presence, on the other hand, do not show a significant difference between any of the variants, since they seem to be perceived mainly with respect to the virtual environment, instead of the handshake mechanism.

While the overall preference given by the rankings and the results from the custom section are likely to give the same outcome with a wider set of tests, the other sections' results could slightly vary with more participants.

## Chapter 7

# Conclusion and future work

The purpose of this thesis work was to explore and develop new ways to handshake that could be integrated in existing Social VR platform, to increment the non-verbal communication available within them and make social interactions more similar to face-to-face communication. The aim was, also, to made them available for these platforms' public, by involving only the controllers and not other additional devices.

After developing four different handshake variants, they have been tested on users to determine which one they preferred in terms of ease of learning, ease of use, fidelity, embodiment, efficiency, social presence, error control and social comfort.

The results of these tests show how, also with the use of controllers, the similarity of movements, that the virtual interaction requires, with respect to the real-world ones is the most relevant characteristic for the users to perceive it as more real, appreciate it more and find it easier to use.

In future works, this analysis could be extended to a system involving the hand-tracking, comparing the absence of haptic feedback, that this method would have, with the discomfort of holding a controller.

Another possible development could be to extend this analysis to other types of interactions, such as high-five and hugs.

# Appendix A

## Pre-experience questionnaire

<b>Età</b>
<b>Genere (M/F/NB)</b>
<b>Professional Status</b> (Bachelor Student, Master student, PhD Student, Post-doc Researcher, Employee (no a student or researcher), other (specify))
<b>What is your English level?</b> 1 (I do not speak nor understand English), 2 (Basic), 3 (intermediate), 4 (Advanced), 5 (Mothertongue)
<b>What is your Italian level?</b> 1 (I do not speak nor understand Italian), 2 (Basic), 3 (intermediate), 4 (Advanced), 5 (Mothertongue)
<b>Do you have experience with programming?</b> 1 (not at all), 5 (A lot)
<b>Have you ever used devices for immersive virtual reality and, if so, how often?</b> (HTC-Vive, Quest2 etc...) 1 (Never), 2 (It may have happened), 3 (Sometimes), 4 (Quite often), 5 (Everyday)
<b>Do you play video games?</b> 1 (Never), 2 (It may have happened), 3 (Sometimes), 4 (Quite often), 5 (Everyday)
<b>Have you ever used Social VR platforms and, if so, how often?</b> (Mozilla Hubs, AltaspaceVR, Vrchat, RecRoom, etc...) 1 (Never), 2 (It may have happened), 3 (Sometimes), 4 (Quite often), 5 (Everyday)
<b>Why do you use these Social VR platforms?</b> 1 (Meet new people) 2 (Keep in touch with friends that live far away) 3 (Keep in touch and socialize with friends in general) 4 (Other) 5 (All of the previous)
<b>For which kind of activity do you usually use SocialVR platforms for?</b> 1 (Chat with other users) 2 (Games) 3 (Virtual Events) 4 (Other) 5 (All the previous)
<b>When you meet a person for the first time, do you usually introduce yourself using the handshake?</b> 1 (Never) - 5 (Always)
<b>Does handshaking people bother you?</b> 1 (Not at all) - 5 (It always does)
<b>Does handshaking people make you feel uncomfortable?</b> 1 (Not at all) - 5 (It always does)

## Appendix B

# Post-variant questionnaire

Embodiment   1 (Strongly disagree) - 4 (neutral) - 7 (Strongly Agree)
Although the virtual body of the <b>avatar</b> did not look like me physically, I felt that the virtual body of the avatar was my own body.
I felt that the movements of the virtual body of the <b>avatar</b> were caused by my own movements.
I felt as if my <i>body</i> was located where I saw the virtual body of the <b>avatar</b> .
I felt "out of my body".

<b>Inconsistence of Presence 1 (Completamente in disaccordo) 5 (Completamente d'accordo) [IPQ]</b>
In the virtual world I had a sense of presence, to "be there" during the experience. (Overall) *
To some extent I felt I was wrapped up in the virtual environment.
I felt like I was just perceiving images (pictures/photos).
I didn't feel present in the virtual environment.
I had the feeling of acting in the virtual environment (from within), instead of operating from the outside.
I felt present in the virtual environment.
How aware were you of the real environment surrounding you as you moved around the virtual world? (sounds, room temperature, other people, etc.) [very conscious - not at all conscious]
I wasn't aware of my real environment.
I still paid attention to the real environment.
I was totally enchanted by the virtual world.
How real did the virtual environment seem to you? [1. totally real - 5. not real]
How consistent was the experience in the virtual world compared to the real one? [inconsistent - very consistent]
How real did you think the virtual environment was? [1. like an imagined world - 5. indistinguishable from a real world]
The virtual world seemed more realistic than the real world.
I was able to correctly perceive the scale, proportions and size of environments/objects in the virtual world. *
When I think about the VR experience, I remember the virtual world as a place where I've been. *
I had the feeling of sharing the virtual world with other people, as if we were really in the same place. *

Figure B.1: \* questions repeated for each variant.

<b>Social Presence   1 (Completamente in disaccordo/mai successo), 7 (Completamente d'accordo/spesso)</b>
I hardly notice the other individual.
The others noticed me in the virtual environment /(are aware of my presence).
I was often aware of others in the environment.
Others were often aware of me in the room.
I think the other individual often felt alone.
I often felt as if I was all alone.
When I was happy, the other was happy.
When the other was happy, I was happy.
The other individual was influenced by my moods.
I was influenced by my partner's moods.
The other's mood did NOT affect my mood/ emotional-state.
My mood did NOT affect the other's mood/ emotional-state.
I think people in this virtual environment affected each other's mood/emotional-state.
I sympathized with the other person.

My actions were dependent on the other's actions. *
The other's actions were dependent on my actions. *
My behavior was in direct response to the other's behavior. *
The behavior of the other was I direct response to my behavior. *
What the other did affected what I did. *
What I did affected what the other did. *
My partner did not help me very much.
I did not help the other very much.
My partner worked with me to complete the task.
I worked with the other individual to complete the task.
The other could not act without me. *
I could not act with the other. *

Figure B.2: \* questions repeated for each variant.



Post-variant questionnaire

<b>Rate these sentences. 1 (Completamente in disaccordo), 5 (Completamente d'accordo) [SUS] (SISTEMA = HANDSHAKE)</b>
I think I could use the system frequently/for long periods to do this task.
I found the system too complex.
I think the system is easy to use.
I think I would need the support of a "technician" to use the system.
I found the various system functions well implemented.
I think there were too many inconsistencies in the system.
I think most people would learn to use a system like this quickly.
I found the system uncomfortable to use.
I felt confident using the system.
I needed to learn many things before I could use the system.
<b>VRUSE 10</b>
I think the system was working against me.
I had no clear idea how to use a particular function.
The overall system response time did not affect my performance.
The system did NOT behave as I expected.
It was hard to learn how to use the system.
I felt in control of the system to perform the task assigned.
I can see a real benefit in this Handshake system.
I enjoyed using this system.
The system was adequate to perform the task assigned.
<i>Overall I rate the usability of the system as: very unsatisfactory - very satisfactory.</i>

Post-variant questionnaire

<b>VRUSE 8</b>
It was easy to undo/recover my errors and go back to a previous state.
I was unaware of making mistakes.
The system kept me from making trivial mistakes.
There was no way to undo an operation.
It was impossible to make silly mistakes.
The VR system was robust and reliable.
<b>VRUSE 1</b>
The features offered by the system were adequate to perform the assigned task.
The features offered by the system were ambiguous.
It was hard to remember all the system features.
I understood how the control system worked.
I didn't need to use all the system features to complete the assigned task.

<b>Custom</b>
It was easy to complete the handshake started by the other person.
It was easy to start/propose the handshake.
It was inevitable to complete the handshake initiated by the other person.
The system was fast/efficient (I could complete many handshakes in no time).
I often had the feeling that I was socializing with another person.
It seemed to me to shake hands with a real person (as if you were not in a virtual environment).
I could shake hands without looking at my hand/controller.
How would you judge the other person's personality (based on the handshake): 1 shy, 2 hesitant, 3 introvert-calm, 4 confident, 5 extroverted
Who had the "pulse" of the situation? Who was leading?: 1 Me - 5 the other person

## Appendix C

# Post-experience questionnaire

Satisfaction [QoE]   1 (Strongly Disagree), 5 (Strongly Agree)
I enjoyed experiencing the virtual environment.
The simulation was easy to use.
I needed to learn a lot of things before I could get going with this system.
I would have liked more time in the virtual environment.
My experience did not meet my expectations.

<b>Rank the variants. [AD HOC]</b>
With which variant was easier to Initiate the Handshake
With which variant was easier to Complete the Handshake (initiated by the other user)
With which variant was easier to Cancel the Handshake Initiated by me
With which variant was easier to Cancel the Handshake Initiated by the other User
With which variant was easier to Avoid/Ignore to Handshake
Which variant was the easier to learn
Which variant was the easier to use
Which variant was felt the most natural
With which variant I felt more in control of the task
Which variant was the least distracting
In which variant i felt the most that the avatar was my body (embodiment)
In which variant i felt the most that the virtual hand were my hands (embodiment)
Which variant was the most efficient (more hanshakes in lesser time)
With which variant i felt more comfortable/at ease
Which variant made me felt I was handshaking a real person
Which variant I preferred overall

# Bibliography

- [1] Danah M. Boyd and Nicole B. Ellison. «Social Network Sites: Definition, History, and Scholarship». In: *Journal of Computer-Mediated Communication* 13 (Oct. 2007), pp. 210–230 (cit. on p. 2).
- [2] Joo Tang-Mui and Chan-Eang Teng. «Impacts of Social Media (Facebook) on Human Communication and Relationships: A View on Behavioral Change and Social Unity.» In: *International Journal of Knowledge Content Development Technology* 7 (Dec. 2017), pp. 27–50 (cit. on p. 2).
- [3] Qiaoxi Liu and Anthony Steed. «Social Virtual Reality Platform Comparison and Evaluation Using a Guided Group Walkthrough Method». In: *Frontiers in Virtual Reality* 2 (May 2021) (cit. on pp. 4, 17, 21, 22, 26, 37).
- [4] Divine Maloney, Guo Freeman, and Donghee Yvette Wohn. «"Talking without a Voice": Understanding Non-verbal Communication in Social Virtual Reality». In: *Proceedings of the ACM on Human-Computer Interaction, Volume 4, Issue CSCW2*. Oct. 2020. URL: <https://dl.acm.org/doi/abs/10.1145/3415246> (visited on 08/20/2022) (cit. on pp. 4, 5, 20, 26, 37).
- [5] Haiyan Wang. «Nonverbal Communication and the Effect on Interpersonal Communication». In: *Asian Social Science* 5 (Oct. 2009) (cit. on p. 5).
- [6] Theresa Jean Tanenbaum, Nazely Hartoonian, and Jeffrey Bryan. «"How do I make this thing smile?": An Inventory of Expressive Nonverbal Communication in Commercial Social Virtual Reality Platforms». In: *CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Apr. 2020. URL: <https://dl.acm.org/doi/10.1145/3313831.3376606> (visited on 10/04/2022) (cit. on pp. 6, 22, 23, 26, 37).

- [7] Edward T. Hall. *The Hidden Dimension*. Doubleday Garden City, N.Y, 1966 (cit. on p. 6).
- [8] Mark L. Knapp, Judith A. Hall, and Terrence G. Horgan. *Nonverbal Communication in Human Interaction*. Cengage Learning, 2013. ISBN: 978-1-133-31159-1. URL: [https://books.google.it/books?id=-g7hkSR\\_mLoC&hl=it&source=gbs\\_book\\_similarbooks](https://books.google.it/books?id=-g7hkSR_mLoC&hl=it&source=gbs_book_similarbooks) (cit. on p. 6).
- [9] *Meaning of virtual reality in English*. URL: <https://dictionary.cambridge.org/dictionary/english/virtual-reality> (visited on 10/08/2022) (cit. on p. 8).
- [10] Nikola Rendeovski, Diana Trajcevska, Mario Dimovski, Konstantin Veljanovski, Aleksandar Popov, Naile Emini, and Dimitar Veljanovski. «PC VR vs Standalone VR Fully-Immersive Applications: History, Technical Aspects and Performance». In: *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)* (June 2022), pp. 1–4 (cit. on p. 14).
- [11] *What Is Social VR, and How Does it Work?* URL: <https://vr.space/news/education/what-is-social-vr/> (visited on 10/18/2022) (cit. on p. 17).
- [12] Vlasios Kasapakis, Elena Dzardanova, Vasiliki Nikolakopoulou, Spyros Vosinakis, Ioannis Xenakis, and Damianos Gavalas. «Social Virtual Reality: Implementing Non-verbal Cues in Remote Synchronous Communication». In: *EuroXR 2021: Virtual Reality and Mixed Reality*. Cham: Springer International Publishing, Nov. 2021, pp. 152–157. URL: [https://link.springer.com/chapter/10.1007/978-3-030-90739-6\\_10](https://link.springer.com/chapter/10.1007/978-3-030-90739-6_10) (visited on 11/02/2022) (cit. on pp. 24, 25, 37).
- [13] Cloe Huesser, Simon Schubiger, and Arzu Çöltekin. «Gesture Interaction in Virtual Reality». In: *Human-Computer Interaction – INTERACT 2021*. Cham: Springer International Publishing, Aug. 2021, pp. 151–160. URL: [https://link.springer.com/chapter/10.1007/978-3-030-85613-7\\_11](https://link.springer.com/chapter/10.1007/978-3-030-85613-7_11) (visited on 11/02/2022) (cit. on pp. 25, 37).
- [14] *XR Interaction Toolkit Documentation*. URL: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html> (visited on 10/27/2022) (cit. on p. 32).

- [15] *Photon PUN 2 Documentation*. URL: <https://www.photonengine.com/pun> (visited on 10/27/2022) (cit. on p. 33).
- [16] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. «"Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics"». In: *CHI '21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. May 2021. URL: <https://dl.acm.org/doi/10.1145/3411764.3445582> (visited on 11/02/2022) (cit. on p. 37).
- [17] *Input Devices for Virtual Reality (VR) Systems*. URL: <https://techooid.com/input-devices-vr> (visited on 10/14/2022).

# Acknowledgements

I would like to thank the professor Fabrizio Lamberti and the professor Adalberto Simeone, who gave me the opportunity to work on this thesis' project and allowed me to live this experience abroad, inviting me in the professor Simeone's team at KU Leuven. In addition, thanks to Dott. F. Gabriele Praticcò for the support and assistance provided to me in the last months of this project.

Special thanks goes to my parents and my sister, who always supported and encouraged me to give my best in my (long) university studies.

Thanks to Leone, who has been supporting and standing me for many years now, in all my outburst and in my madness (positive and negative).

A huge thanks goes to every person I met during my university career, from who I've been with just for a few months, to who I spent years with.

In particular, I thank Fabio (without whom this project and many others wouldn't have been possible), Federica, Elisa and Cesare, that between group projects, movie marathons, laughs and support calls from kilometers of distance, have been my life mates of the last two years.

I'd like to thank Roberta and Chiara, for moral and psychological support during these months of thesis, because without them I would probably still be waiting locked outside of the lab, I would have had lunch by myself for weeks and I would have had a mental breakdown every second.



### *Acknowledgements*

---

Thanks to my best friends Giulia and Miriam. They have been by my side since, respectively, elementary school and high school and they managed to cheer me up and root for me all this time, giving me the confidence I have always lacked of.

Last, but not least, thanks to my "soubrette", Sofia, Camilla, Roxana, Angelica, Rachele and Elisa. I have spent with them every single day of my stay in Belgium and, during those months, they have become a second family to me and my support system during the hard times there.

I will bring a piece of you with me in every adventure that life will give me.