FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Mobile phone as VR gateway

Davide António Ferreira Castro

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Rui Pedro Amaral Rodrigues

Second Supervisor: Prof. Teresa Carla de Canha e Matos

July 28, 2023

Mobile phone as VR gateway

Davide António Ferreira Castro

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Daniel Filipe Martins Tavares Mendes
Referee: Prof. Telmo Miguel Oliveira Adão
Referee: Prof. Rui Pedro Amaral Rodrigues

July 28, 2023

Resumo

A Realidade Virtual (RV) é um assunto que recentemente tem tido grande foco nos contextos de investigação científica e desenvolvimento de aplicações. Realidade Virtual permite a criação de mundos inteiramente virtuais, incluindo ambientes, objetos, pontos de interação e eventos. No entanto, o principal foco de RV, e também a sua maior vantagem, é a habilidade de poder manter utilizadores imersos nestes mundos da forma mais natural possível.

Dentro de um mundo de Realidade Virtual, um utilizador pode ser alvo de estimulação visual e auditiva através de ecrãs e *headsets* de tecnologia de ponta, e vários dispositivos foram desenvolvidos para os utilizadores interagirem e controlarem o ambiente virtual em sua volta de uma forma acessível. Embora permita imersão e controlo sem precedentes, a maior parte das abordagens exploradas neste contexto limitam as capacidades de o utilizador interagir com o mundo real, incluindo a interação com os seus dispositivos reais.

O telemóvel é um objeto destacadamente presente no dia-a-dia de grande parte das pessoas e impactou significativamente a forma como os seus utilizadores vêem o mundo e como interagem com ele. Desde interações sociais a comunicação no contexto laboral, o telemóvel fornece um grande número de funcionalidades usadas com regularidade durante o dia pelo utilizador comum. No entanto, a barreira entre o mundo virtual e o real que é imposta pela tecnologia da Realidade Virtual torna o telemóvel difícil, e por vezes impossível, de usar sem interromper a experiência. Com o surgir do desejo de incluir objetos e dispositivos como o telemóvel em RV, o conceito de Virtualidade Aumentada (VA) foi criado. O foco da VA é de transportar objetos reais para ambientes virtuais de Realidade Virtual.

Com esta dissertação, e como resposta ao problema apresentado, exploramos as diferentes técnicas que podem ser aplicadas para integrar o telemóvel, juntamente com as suas funcionalidades mais importantes, dentro de um ambiente virtual. Os aspetos cruciais da integração foram alvos de investigação, incluindo a representação visual e a interação do utilizador com o dispositivo. Através do desenvolvimento de um protótipo funcional, nós aplicamos um conjunto de técnicas, obtido através da investigação efetuada, para fornecer uma experiência de Virtualidade Aumentada de utilizar um telemóvel num ambiente virtual.

Foram conduzidos testes com utilizadores no protótipo para avaliar o seu desempenho e responsividade e aferir a sua viabilidade em manter uma experiência acessível e adequada. Os resultados do processo de avaliação mostraram que os utilizadores foram recetivos à ideia de usar o telemóvel em ambientes de RV e conseguiram executar tarefas simples e essenciais sem problemas significativos.

Palavras-chave: Realidade Virtual, Virtualidade Aumentada, telemóvel

Abstract

Virtual Reality (VR) has been the subject of much research and development of all kinds of solutions in recent years. It allows for the creation of completely virtual worlds, including environments, objects, interaction points, and events. However, the real focus and advantage of VR is the ability to immerse users in these worlds in the most natural way possible.

Within a Virtual Reality world, the user can receive visual and auditory stimulation through state-of-the-art head-mounted displays and headsets, and devices have been developed for the user to interact and control the virtual environment in an accessible manner. Although allowing for unprecedented immersion and control, most devices and approaches that are used limit users' capabilities to interact with the real world, including their everyday devices.

As an item prominently present in most people's everyday lives, the mobile phone has significantly impacted how its users perceive and interact with the world. From social interactions to workplace communication, the mobile phone provides a large set of features used regularly during the day by the average user. However, the barrier between the virtual and real worlds imposed by VR technology makes using the phone difficult and even unfeasible without interrupting the VR experience. As the desire to include objects and devices, such as the phone, in VR appeared, the concept of Augmented Virtuality (AV) was created. The focus of AV is to accurately transport real-world elements into virtual reality environments.

With this dissertation, as a response to the problem presented, we explored the different techniques that can be applied to integrate the mobile phone and its most important features in the virtual environment. We approached the most crucial aspects of the integration, including the visual representation and user interaction with the device. Through the development of a working prototype, we applied a set of techniques, obtained through all the research done, to provide an Augmented Virtuality experience of operating a mobile phone in a virtual environment.

User experiments were conducted on the prototype to evaluate its performance and responsiveness and assess its viability in keeping a user-friendly, accessible and adequate experience. The evaluation results showed that users were receptive to the idea of accessing their phones in VR environments and could perform simple and essential talks without significant problems.

Keywords: Virtual Reality, Augmented Virtuality, mobile phone

Acknowledgments

Firstly, I want to thank my supervisors, Prof. Rui Rodrigues and Prof. Teresa Matos, for their patience and guidance. Their constant availability was an essential support throughout the year, and their dedication and encouragement never failed to make me push myself to improve.

I want also to thank everyone at the GIG laboratory, for the welcoming and supportive environment provided to me throughout my work, and also FEUP and DEI for the necessary spaces and equipment for the development of this dissertation.

I express my gratitude to everyone who participated in the user tests for their presence and feedback.

To my friends and colleagues, a big thank you for their support and always helping me find a light of fun and laughter among all the problems and difficulties. To my lab friends Diogo and Henrique, for helping me throughout this work with their presence, clever contributions and undying sense of humour.

Last, but definitely not least, I thank my family for always being there for me, ever since I can remember, and pushing through all the hard times by my side. Thank you for all the love and support you have given me throughout all these years. Everything I am and have accomplished would not have been possible without you.

Davide António Ferreira Castro

“The noblest pleasure is the joy of understanding.”

Leonardo da Vinci

Contents

List of Figures

List of Tables

List of Code Snippets

Abbreviations

AV	Augmented Virtuality
HMD	Head-Mounted Display
IR	Infrared
UCD	User-Centered Design
VR	Virtual Reality

Chapter 1

Introduction

In recent years, Virtual Reality (VR) technology has grown constantly, showing its potential for many uses. From social interaction to enhanced entertainment and virtual workplaces, this technology allows users to perform many activities in a completely virtual and immersive environment. With this in mind, the isolation from reality one experiences in the virtual environment is a possibly vital factor to consider.

1.1 Context and Motivation

Advancements in Virtual Reality technology led to the research and development of many applications, which pushed the market forward and attracted users from different backgrounds and occupations.

The main focus and attraction of VR are that it is a unique way for users to place themselves in a world that, essentially, does not exist. Creators and users can manipulate this world in a virtually infinite number of ways, only constrained mostly by computational restrictions, content creation, and human labour. Therefore, this technology can be used to simulate or visualize environments, objects, and scenarios that would otherwise be impossible or extremely hard to recreate in the real world. Nowadays, it is used in various areas, such as tourism, architecture, design planning and collaboration, safety training, human behaviour and perception [31], mental health studies, education, and much more.

Although, as stated before, recent innovations in these virtual environments and applications have many possibilities and advantages, VR users depend on other factors that are only present in the real world. Regarding this, we can think of the tools, objects, and devices users operate regularly throughout the day.

The concept of Augmented Virtuality (AV) was introduced to define this arising desire to include real-world environments and objects, including mobile devices, in virtual reality applications. This consists of embedding real-world objects in the virtual environment [10]. A number of different possibilities have been researched and developed with a focus on AV and its impact on various areas.

1.2 Problem

Immersion into a virtual environment, using the latest VR devices such as head-mounted displays (HMDs), usually implies sacrificing the user's connection to the real environment around them. While using an HMD, the user cannot see the outside world, sometimes not even hear it, and, while holding the controllers and movement restrained, cannot appropriately interact with their surroundings.

As helpful as a virtual environment can be, there is currently a barrier that alienates the user from their reality, that is, their everyday affairs, connections, and possessions, proving to be an inconvenience and even an isolating factor in their life. This disconnection between what can be used in VR and what can be used in reality can be improved by providing the user with an accessible gateway to their real devices within the virtual environment [20].

The use of the mobile phone - which has taken its place as the primary device in the average person's life - inside the virtual environment would provide a unique and powerful opportunity to make the most of the device's features while taking advantage of the virtual reality experience. However, integrating such devices inside a virtual environment presents a complex challenge due to their interaction possibilities, number of features, and physical connection to the real world.

1.3 Research Questions

To use the mobile phone from within a virtual experience to access its features and information, some aspects should be considered, such as the audiovisual representation of the device in the generated environment, the means of interaction, and meaningful feedback. Based on the problem mentioned, the following research questions were set:

1. Which mobile phone tasks are possible to do in a VR environment?

2. What techniques may allow a user to operate and complete tasks with the mobile device in the virtual environment with efficiency as close as possible to reality?

1.4 Objectives

This work aims to study how to provide an efficient and reliable way for a Virtual Reality user to operate their mobile phone, with as many of its functionalities as possible, from within a virtual environment. By exploring different techniques previously used, not necessarily in this specific context, we tried to determine how to combine them and improve them to integrate all the components of the mobile device into the virtual world. Therefore, we aimed to identify the most critical features of the mobile phone and how to integrate them while still taking advantage of the VR capabilities. We also intended to ensure the best approaches to replicate the mobile device as accurately as possible within virtual worlds.

With these objectives in mind, and in order to achieve them, we outlined a set of tasks that were executed throughout this work:

- Research on the current techniques established to approach Augmented Virtuality through mobile devices
- Identify the best approaches to display the mobile phone inside the virtual environment
- Apply AV techniques to enhance the interaction with the mobile phone
- Make conclusions on which features and strategies work through user testing

1.5 Document Structure

In chapter 2, we discuss the state of the art concerning the background and context of the problem, presenting the research and works previously done around the main relevant concepts, covering the current developments on VR interaction, VR with mobile devices, and integration of the mobile phone in VR environments. Chapter 3 presents the proposed solution to the problem, including the requirements and architecture. In chapter 4, we discuss the implementation of the prototype, developed according to the proposed solution, approaching the process regarding the various components of the prototype. The evaluation process is addressed in chapter 5, where we discuss the methodology used for the testing of the prototype and present a deep analysis of the resulting data. Finally, in chapter 6, we present the conclusions taken regarding the information obtained and presented in the previous chapters, as well as what future developments and improvements can be done for this work.

Chapter 2

State of the Art

To better contextualize the work done, this section includes definitions of the most important concepts surrounding Virtual Reality and the use of mobile devices, while covering and referencing previous works related to this topic.

In section 2.1 the primary concepts related to Virtual Reality and Augmented Virtuality are explored, providing a background on the current technological state of possible interaction methods and techniques in virtual environments. Section 2.2 focuses on the mobile phone and its relevance in the context of VR, divided into three subsections. Firstly, a rundown on its features and applications (Section 2.2.1), followed by research on how it has previously been used in VR and for what end (Section 2.2.2). Finally, in Section 2.2.3 we discuss the AV-focused techniques that have been developed and explored in previous works to properly integrate the mobile phone into a virtual environment.

2.1 VR and Augmented Virtuality

The research and work performed around Virtual Reality (VR) have constantly increased in recent years as available technology improves and different new possibilities inside the virtual world are explored. However, although the realization of a purely virtual environment for users to see and interact in has been vastly explored throughout the years, the desire to include elements of reality in the virtual world, and vice-versa, has also become a research subject. In fact, the concept of mixing reality and virtuality, referred to as "mixed reality", has been a frequent subject of research projects. Milgram and Kishino [21] have proposed the virtual continuum, represented in Figure 2.1, which represents the spectrum between the real or physical environment and the digital or virtual environment. It is defined that, between full reality and full virtuality, there are two categories of mixed reality: Augmented Reality (AR) and Augmented Virtuality (AV). While AR focuses on integrating virtual objects into the real world, for example, by using a camera and displaying 3D-rendered objects, AV is the exact opposite. Augmented Virtuality is defined as “to have a ‘window’ inside the virtual environments to see the outside real world” [26]. Therefore,

Figure 2.1: Representation of the virtuality continuum (Milgram Diagram) [21]

as we want to bring the real-world object, in this case, the mobile phone, into the virtual reality environment, we see that the AV is at the core of what is intended from this work.

The inclusion of a visual representation of the mobile phone inside the virtual environment, however, is not our only focus. Having in mind the device's features and means of use, the capability of interaction is just as important.

Interaction in VR

Apart from the head-mounted display (HMD), arguably the most essential device developed for VR experiences, many other devices, approaches, and techniques have been explored for the interface between the user and the virtual environment. One of the most popular means of interaction in VR is the use of a controller. Despite the many variations of controllers for VR, they usually consist of a pair of hand-held devices of easy motion that support some form of input from the user. Usually, the controller has a few buttons for the user's thumb, a trigger button on the back for simple actions such as grasping elements of the virtual environment, and sometimes a touchpad and a thumbstick. Some examples of these traditional controllers are the HTC Vive, portrayed in figure 2.2, the Meta Quest 2, and the Samsung HMD Odyssey+ controller [4]. These controllers allow for relatively precise interaction in the virtual environment and, as the position of the device is tracked, can be helpful in many scenarios. Being the most common interaction device for VR, they are used for most basic and essential actions. Some of these include movement through the virtual world using, for example, the thumbstick, and simple contextual inputs using the buttons. Also for selection, such as in menus or other interfaces, it is possible to use touchpad or, optionally, use the controller as a pointer through its tracking capabilities. Some controllers can go even further, as is the case with the Valve Index controllers which support the calculation of the pose of the user's hands and individual fingers. The use of this data provides an advantage in interaction possibilities compared to competitors' controllers and, in turn, providing an experience closer to reality [4].

However, although mostly practical, these devices provide relatively low interaction flexibility. Users are limited to interactions with their hands and, apart from the controllers that use finger detection and pose tracking, limited to the actions that are possible using the traditional buttons and thumbsticks. Additionally, the user must be holding the controller at all times, limiting what they can do with their hands. Perhaps even more important is the fact that these devices provide a more unnatural means of interaction [26]. In fact, VR users find themselves using purposely

Figure 2.2: HTC Vive for XR Series

crafted physical devices with controls and input methods that might change in between contexts and that the user has to learn.

In response to controllers' limitations, solutions for more natural means of interaction have also recently been researched and developed. Looking to go beyond controllers' hand and finger posture tracking systems mentioned previously, additional hand tracking solutions were explored. These systems are capable of higher precision that is needed for the significant complexity of hand movements and gestures. An example is the data gloves, such as the CyberTouch [1] device, portrayed in figure 2.3, which can collect information on the posture of the user's hands and the movement of the individual fingers [15]. However, the use of data gloves implies the attachment of extra hardware to the user's hands, such as the gloves and the sensors on the fingers. As this might constrain the hand gestures and provide an uncomfortable experience for the user, other solutions without the need for attached hardware were developed, such as the Leap Motion Controller. Only requiring a small optical device, the Leap Motion can recognize hand gestures with high accuracy and low latency through a support vector machine (SVM) [15].

Another solution for natural interaction is full body tracking, for which there are two main options. Firstly, there are systems that are capable of performing very precise capture of body movements, known as motion capture systems. This usually requires the use of motion suits that have markers attached, as portrayed in Figure 2.4, that can be used to calculate the body posture in real-time using infrared cameras. Alternatively, inertial sensors can be used instead of markers. These sensors are composed of accelometers and gyroscopes, which allow to calculate the movement and rotation of every part of the body they are placed on [15]. Due to the resources needed for this method of body tracking, it is a very expensive option and is often only used for large-budget projects, such as films and video games. The other option consists of using infrared depth cameras, such as Microsoft's Kinect, as can be seen in Liu et al. [17] where a Kinect device is used to track body posture and generate a virtual skeleton. The infrared images allow the device to produce a 3D image of the objects in the camera's field of view and, based on that

---

[1] CyberTouch, `http://www.cyberglovesystems.com/`, (accessed 04/07/2023)

Figure 2.3: Hand tracking with data gloves from CyberTouch

information, attempt to recognize the structure of a human body and, consequently, its posture and movements. This allows for the recognition and processing of input directly from the natural body movement of the user to interact with the virtual world. This system has the benefit of requiring very few hardware elements, which also do not have to be directly controlled or carried by the user, increasing the flexibility of the possible set of gestures and, therefore, decreasing restrictions.

Additionally, more recent VR systems also include support for other complex interactions, such as eye-tracking. By incorporating eye trackers inside the VR headset it is possible to track the user's gaze direction at all times. This can be used to, for example, provide a quick and easy way for users to select objects in the virtual world and, with the inclusion of head gesture tracking, perform actions upon selections [23]. For example, Biener et al. [7], in order to develop a virtual offices for workers, used eye-tracking to allow for easier selection of screens and windows.

All interaction systems mentioned above, however, require additional specific hardware to work, which may represent an obstacle for most users. Having this in mind, another alternative that has also been considered for interaction in VR is the mobile phone. Although the previously mentioned systems might provide more natural means of interaction, the mobile phone can also provide an easy and familiar experience, taking advantage of its many functionalities.

2.2 Mobile devices and VR

As the mobile phone device is present in most users' daily routines and, therefore, subject to regular use, its operation as an interaction device within the virtual environment could feel more familiar than a traditional VR controller. The fact that the users already know how the phone works and most likely have already learned how to use its features and input hardware makes the learning process much more manageable. And as it is a device that most users already own and does not require any additional hardware to work, it represents no extra costs on software and hardware. This represents an obvious advantage compared to the other interaction options mentioned previously, as most interaction and tracking methods required additional hardware beyond the HMD, such as, for example, the depth cameras for body and hand posture tracking. Additionally, the mobile

Figure 2.4: Setup of motion capture suit

phone also provides many other features beyond input, such as providing output information to the user and allowing the use of applications for many different purposes.

2.2.1 Smartphone applications and features

After many years of advancements in the smartphone industry, these devices can now provide many different services to its user. From the most simple use of the device, for phone calls and text messages, to the use of sophisticated software in the form of applications, the smartphone is a constant presence in the everyday life of most of the world's population[2].

The smartphone is, for most people, one of the primary means of obtaining information and managing various life aspects, such as work and social relationships, through the internet. In fact, reportedly 92.3 percent of internet users access it through their smartphone, representing 56.9 percent of the total online time[3]. Nowadays, many applications allow users to easily share content with others, send messages, and perform voice and video calls.

Using a smartphone affects the user's social and professional life and, consequently, their psychological health. The instant access to a way of interacting with friends and family at all times helps reduce work stress and increase social support [24]. A study by Chan on the relationship between the mobile phone and social capital [8] also shows that mobile online communication is related to bridging social capital, with even voice communication being stronger. The device is also beneficial for users to organise their schedules and to-do tasks throughout the day, whether related to their job or personal life. There are applications for easy scheduling of events and reminders on the calendar, as well as for mental maps and notes that are more convenient than traditional methods and can be helpful in many contexts in the user's everyday life.

---

[2] Number of smartphone subscriptions worldwide from 2016 to 2021, with forecasts from 2022 to 2027, `https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/`, (accessed 04/07/2023)

[3] Digital around the world, `https://datareportal.com/global-digital-overview`, (accessed 04/07/2023)

We can see that there are currently many features that, when aiming to integrate the smartphone into another medium, would be crucial to include due to their relevance. Among them is the ability to receive and make phone calls, as well as the use of messaging and social media applications. Additionally, the smartphone supports applications with strong connections to reality, such as location and orientation services and the camera.

2.2.2 Smartphone in VR

The vast number of functionalities and sensors in mobile devices, as well as their easy portability, have been a big incentive for developing VR solutions which try to bring some of the advantages of these devices to the VR experience.

Some solutions have attempted to improve the access to information from the mobile device that may be useful to the user and include data from the phone into the VR experience. To illustrate this, we have the notifications' solution in Meta Quest devices. With these devices, it is possible to link a mobile device to the VR device using a mobile application and, when there is a new notification on the phone's lock screen, it will be displayed to the user through a heads-up display inside the virtual environment [4]. However, this solution is very simple and limited only to displaying phone notifications, lacking the ability to provide any interaction with the mobile device.

Alternatively, others have explored the potential of the mobile phone to become an interaction device in VR. One of today's mobile phones' main components is the frontal panel touchscreen. With the precise reading of touch input, including multi-touch support for multiple contact points simultaneously, the touchscreen provides a set of different uses as an interaction device in a virtual environment. The integrated touchscreen of the mobile phone has been used in previous works as an interactive surface for tasks such as manipulating objects in the environment [9, 25], navigating across the virtual environment [11] and even interacting with menus, graphical interfaces and others within the virtual world, as in Lipari et al. Handymenu [16].

Another relevant task within virtual environments is text input, which can be challenging due to the impracticality of using a physical keyboard, the most common means of text input, in a VR setting. The user would always need a surface to place the keyboard, such as a desk, and accurate input becomes difficult without seeing the keyboard and fingers inside the virtual world. Previous works have therefore tried to take advantage of the portability and touch capabilities of the mobile phone to provide a more efficient solution for text input in VR. To illustrate, Kim et al. [12] present a method, by the name of HoVR-Type, to include in the virtual world a visual representation of a keyboard, which the user can interact with using their phone's touchscreen, as demonstrated in figure 2.5. Additionally, the keyboard gives visual feedback when the user's fingers hover over the keys for better and more accurate input. However, to be able to detect the fingers hovering over the touchscreen, only smartphones that have hover detection support can be used. For example, for HoVR-Type a Samsung Galaxy S4 was used, which supported a "floating touch" feature, intended

---

[4] Phone notifications on Meta Quest headsets, https://www.meta.com/help/quest/articles/in-vr-experiences/social-features-and-sharing/phone-notifications/, (accessed 04/07/2023)

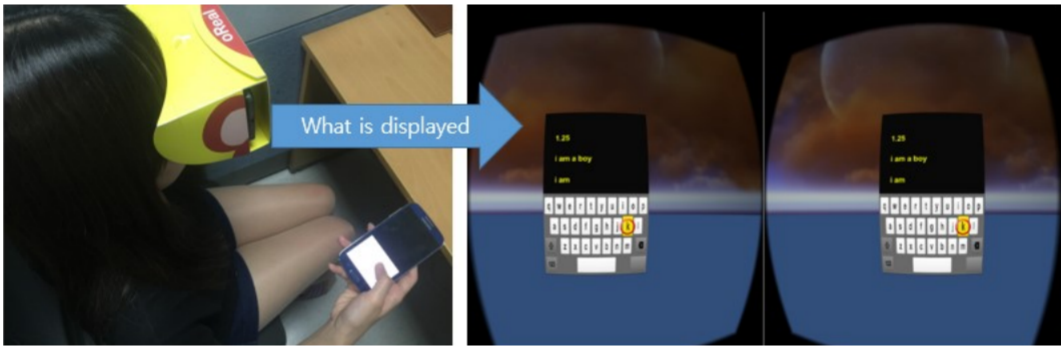


Figure 2.5: Virtual keyboard and feedback, presented by Kim et al. [12]

for uses such as showing information on screen when the user hovered their finger over elements in the screen, by the name of Air View[5]. The touchscreen in that smartphone included capacitive sensors that could accurately detect fingers further away from them. Smartphone touchscreens with support for such features were, however, discontinued.

Considering the capabilities of other sensors integrated into the mobile phone, such as the accelerometer and gyroscope, others have also explored the device's capabilities as input for movement, including position, translation and rotation. This allows for various uses, such as controlling navigation in the virtual world in a more intuitive manner [11, 6] and as an alternative for keyboards and mouses in different situations [13]. For example, Tan et al. [6] explore steer based control to easily simulate metaphors such as walking and airplane navigation, using the rotation of the device (heading, pitch and roll).

2.2.3 Technical aspects in mobile phone VR integration

The use of the smartphone, precisely as it is in reality, within the virtual reality environment, has been a subject of research in several previous works. In fact, as the incorporation of these mobile devices is this work's central point, it is important to analyze the different parts of its process.

Three main conditions should be met to better integrate the mobile device in virtual reality. Firstly, the user must be able to see, without having to remove the HMD, the display of the device's screen in the virtual world. Secondly, the user must have some way of interacting with the phone, specifically with the primary means of interaction, the touchscreen. Lastly, the virtual position of the device should accurately follow the corresponding position relative to the user in reality. Additionally, we can explore other aspects that further enhance the AV (Augmented Virtuality) experience by bringing more elements of reality into the virtual world beyond the device.

---

[5] Samsung Air View Finger, `https://www.samsung.com/hk_en/support/mobile-devices/what-is-the-air-view-feature-and-how-do-i-use-the-air-view-feature/`, (accessed 04/07/2023)

Seeing the device's display in the virtual world - Screencasting

The first step to bringing a mobile device, such as the smartphone, into an entirely virtual and computer-generated world is to break the visual boundaries imposed by virtual reality. When someone uses an HMD (head-mounted display) for virtual reality, they become essentially blind to their physical surroundings. However, the user must be able to see what is displayed on the screen of the device, to correctly operate it and make use of most of its features. For this reason, many approaches to solving this problem were explored in past works.

One technique used for casting the screen of the device is implementing a server-client communication to transmit screenshots from the mobile phone to the virtual reality application. Desai et al. [10] and Alaee et al. [2], for example, realized this idea by building an Android application that serves as a TCP/IP server and sends screenshots at a chosen frequency to the VR application, designed using the Unity game engine. As the screenshot functionality is already built into the Android framework, only the development of a server app was needed. We can see that this approach is then simple and effectively mirrors the screen in the virtual world. However, as stated in Shin et al. [26], the approach always seemed to introduce noticeable delay, as screenshots, being image captures of the screen, represented a considerable amount of data that had to be gathered and sent over the network connection.

The delay or latency of the screen mirroring is a very important factor to have in mind. The user will be interacting with the device, using the visual representation of the device in the virtual world as a reference. Therefore, if the delay between the interaction and the feedback display is too large, the usage performance of the device is potentially compromised. To fix the delay in previous implementations, Kyian et al. [14] explored the possibility of using real-time communication (RTC) by developing an application that, using the WebRTC protocol, transmitted the screen to the VR application. According to their studies, it successfully outperformed the previous screenshot stream approaches.

Interacting with the mobile phone - Touchscreen mapping

The mobile phone's touchscreen, as mentioned previously, is the main contact point between the phone and its user and, therefore, the most used mean of interaction with the device's features. It is capable of detecting movement gestures, such as swiping, and non-movement gestures, such as selecting a point on the screen, and can usually also detect gestures with multiple contact points.

The use of the event data captured by the touchscreen is critical for applications that intend to use the smartphone in VR as an input surface for selection or text typing. Son et al. [27] and Kyian et al. [14] are examples that illustrate this. The latter also explores the possibility of indirect input, that is displaying the virtual surface of the touchscreen in a different manner from that of the real screen. For example, the virtual surface can be larger than the real physical size of the surface or the virtual screen can be moved to another position in the virtual environment that could be more convenient for the user to better see the displayed contents.

The applications running in a smartphone, through the underlying operating system, usually use the raw data from the touchscreen as the input to perform most of the possible actions. Therefore, in order to use the device's applications inside a virtual environment while using the integrated touchscreen, the only data that needs to be transferred to the VR application is the output from the system, such as the audio and the content rendered on the display from the applications. However, the data from the touchscreen may also be useful to provide visual feedback to the user in the virtual scene of what gestures are being captured.

Seeing as the user, while in the virtual environment, is disconnected visually from their own hands and the device itself, visual feedback on the contact points and gestures from within the virtual scene would be advantageous to improve the connection and performance of handling the device.

The touchscreen data is usually available through the API provided by the operating system, such as in the Android Framework, for example. These readings can then simply be sent through a chosen means of communication from the mobile phone to a VR application, that can use them to map them into the virtual mirroring of the display.

Another approach that has been explored is the possibility of providing input to the touchscreen without using it directly, that is, without the use of the fingers on the actual touchscreen. This allows for the use of a proxy object that is replaced in the virtual environment by the visual representation of the real device. To this end, Takashina et al. [28] proposed the Quasi-Touch device, portrayed in figure 2.6, that triggers AC signals to activate the capacitive sensors that form the touchscreen. These signals are triggered when the user selects a point in the virtual screen by using, for example, a VR controller, which can also provide haptic feedback upon touch, and mapped into the corresponding position on the physical screen. This approach, however, removes the advantage of the user's familiarity with the use of the physical mobile phone, since the device is not handled directly by the user. Also, other inputs apart from the touchscreen, such as side buttons, fingerprint sensor and microphone would become practically unusable and, therefore, additional components would have to be designed to accommodate their use.

Position of the device in the virtual environment - Position tracking

As mentioned previously, one of the aspects to have in mind when integrating real devices into the virtual environment is the accuracy of the virtual position of the object in relation to its physical position in the real world. This is necessary for the user to effectively fall under the impression that their device has been completely transferred into the virtual environment they are in and, therefore, take advantage of the familiarity that comes with its use. Suppose the device was to be static or habit an arbitrary position in the 3D world. In that case, the user loses all control over the position of the device and, in turn, sees their ability to use it limited. For example, the user would not be able to rotate the device for applications that require landscape mode, and additional input is needed to modify the orientation. As an alternative, an additional device, such as a traditional controller, could be used to control the device's position within the virtual environment by, for

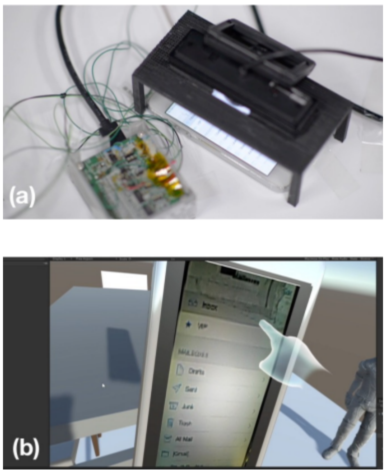


Figure 2.6: Quasi-Touch [28] - Real smartphone with signal emitter (a) and receiver proxy object (b)

example, using the touchpad or the thumbsticks as translation and rotation inputs. However, the necessity to use the mobile phone's touchscreen simultaneously renders this option too impractical. Furthermore, it would increase the feeling of disconnection between the actual device and its virtual representation.

The most common solution to this problem, adopted in previous related works, is to use a VR controller's position tracking capabilities for the mobile phone's position. Amano et al. [3] opted to attach the smartphone to a VR controller included in the HTC Vive Virtual Reality kit. With 6 degrees of freedom (6DoF) tracking, these controllers allow for high precision and performance position tracking. We can also consider the case in the project of Bai et al. [5], where they disassembled a controller from the Meta (formerly Oculus) system kit and modified it to include easy accommodation for the mobile device and still functional tracking. Although this approach yields positive results in performance, based on the mentioned related works, the use of a controller attached to the device is not optimal. In fact, the extra device comes at the price of an unwanted increase in the complexity of handling the device. Apart from the additional work and cost of the setup, the device becomes unavoidably larger and heavier, potentially impacting the user experience negatively.

Another way to reliably track the device's position is to use a fiducial marker, always displayed on the physical screen. As proposed previously by Kyian et al. [14] and Makhsadov et al. [18], the fiducial marker, with the use of a camera attached to the HMD, can be used to successfully perform a 6DoF tracking of the device, without the need for additional sensors. Some HMD models, such as the Meta Quest 2, even have built-in cameras that can be used instead of an external camera. Nevertheless, the marker should be displayed at all times on the screen, which would interfere with the visibility of the rest of the on-screen content at the time of use.

Figure 2.7: A Smartphone fixed onto a VR Controller [3]

Improving the reality-virtuality bridge - AV enhancements

Although the representation of the device itself inside the virtual environment is crucial in the context of this work, other factors can also significantly impact the integration of the mobile device. In reality, a mobile device user relies on one major element to operate it correctly: their hands. For this reason, an inaccurate display of the user's hands and fingers, or the complete lack of it, causes inevitable performance differences compared to the regular use of the mobile device. To illustrate, if a user intends to touch a small specific element of the screen, it becomes easier to precisely to perform touches without fail if the user can have a clear sight of the position of their hands and fingers. Afonso et al. [1] approached this by conducting a study comparing selection performance without virtual hands, a realistic hand and a translucent hand. In fact, they concluded that the presence of a virtual representation of the hands, whether realistic or not, seemed to reduce the occurrence of selection errors.

Matulic et al. [19] explore this theme with the "Phonetroller" system. Facing the problem of touch imprecision due to the user's inability to see their fingers and hands in VR, they explored the possibility of displaying a shadow of the user's thumbs on the virtual screen. In this approach, the front camera of the mobile device and a mirror facing it, attached to the device, are used to constantly capture a front view of the device, as is represented in Figure 2.8. By filtering the user's fingers using image processing, the fingers can be displayed on top of the screen, allowing the user to better visualize the position of their thumbs. However, at the core of this system is the assumption that the real screen of the device will not be used, and it may, instead, be used to provide a green screen and markers for screen edge detection, required for the used techniques of image processing. Additionally, this system requires adding physical components to the device, potentially hindering its handling.

Alternatively, other works have considered using more complex algorithms and equipment. Among others, Alaee et al. [2], Bai et al. [5], and Zhang et al. [30] propose the use of an RGB-Depth sensor. By mounting the depth camera on the HMD, they were able to produce the segmented image of the user's hands. Using the colour and depth values from the camera, it is even possible to build a 3D point-cloud representation of the hands in the virtual world. The Ultraleap hand tracking[6] follows this concept, providing accurate tracking of hand gestures

---

[6] Ultraleap hand tracking, `https://www.ultraleap.com/tracking/`, (accessed 04/07/2023)

Figure 2.8: Thumb display setup [19]

through a head-mounted sensor, the Leap Motion Controller. Other alternatives, using just an RGB camera, are also available, although the lack of depth data limits the accuracy and performance of the algorithm. For example, some HTC Vive headsets use the HMD's RGB cameras to calculate the position and gestures of the hands and provide the 3D data through the Vive SDK[7]. In an attempt to extend this idea even further, a project by the name of "Never Blind VR" [22] made use of a Kinect sensor to display a 3D representation of the user's surroundings in the virtual environment, including their body, other people, and objects nearby.

2.3 Summary

In this chapter, we explored the different fields related to the problem of using the mobile phone from within a virtual environment. The main concepts behind Virtual Reality and Mixed Reality, focused on Augmented Virtuality, were approached, as well as the interaction methods and techniques in VR, the features of the mobile phone and in what ways it has been used in VR, and, finally, the work that has been done on integrating the mobile phone into the virtual world.

We have seen that an extensive set of interaction methods for VR have been developed throughout the years, including the traditional VR controller, which is arguably the most common. Yet, many alternatives beyond the traditional controllers appeared for more natural interaction. However, these options often come with restrictions, such as the costs of additional software and hardware. This, and the number of useful features and applications available within a mobile device, further justifies the research on this device's integration into the virtual world, which is the focus of this work.

The use of the mobile phone for various purposes beyond regular phone utilization has also been explored in this chapter. Due to the variety of sensors integrated into mobile phones, it proved to be a capable device for activities such as spatial navigation, text typing, and an alternative for

---

[7] HTC Vive hand tracking, `https://www.vive.com/eu/support/focus3/category_howto/hand-tracking.html`, (accessed 04/07/2023)

Figure 2.9: AV solution in Alaee et al. [2]. On the left, is the setup with the RGB-Depth sensor. On the right, the user's hands and phone as rendered in the virtual environment.

mouse input. Some of the works related to this subject proved to be of relevant use in the context of this work. For example, the challenges behind implementing a virtual keyboard using the mobile phone's touchscreen revealed the importance of user feedback regarding the position of the fingers and hands for accurate interaction with the touchscreen.

Finally, we considered the work done on integrating the mobile phone along with its set of features in the virtual environment by essentially creating its virtual mirrored version. To this end, we describe and analyze the techniques developed and explored for the various aspects of the technical integration, such as the mirroring of the screen, the mapping of the touchscreen data, the tracking of the device's position, and the integration of other reality elements into the virtual scene, following the concept of AV, such as the user's hands.

Chapter 3

Mobile phone in VR: a proposed solution

After extensive research of state of the art, we tackle the problem previously described once again in light of the background and context of this work, as well as the information and knowledge obtained from the research. This allows us to elaborate on a solution to this problem.

The solution envisioned for this work is a system that allows a VR user to use the mobile phone from within the virtual experience. This directly addresses the problem we explored regarding the disconnection of the user from their surroundings and reality. The use of the mobile phone while using VR equipment and being, in fact, inside a VR experience provides a solid means of interaction with the user's everyday tasks, interpersonal connections, and many more aspects of their life.

As such, the solution we propose provides a representation of a smartphone inside the virtual world, including the phone's display and content, its real position relative to the user, and a generic 3D representation of its body. It also allows physical interaction with the device while still inside a VR environment. We aimed to construct a framework that can support various components of the experience of using a mobile phone. This includes the mobile phone's visual, physical, interactive, and informative aspects. While a virtual copy (mirror) of the phone's screen in VR 3D space might provide the visual content and information in the device, it lacks the interactive and physical nature of the everyday use of the smartphone. On the other hand, using the physical phone as a touchscreen or proxy device for a virtually simulated mobile system may not keep the system's integrity that the users are accustomed to, including their data, such as applications, files, and messages.

Our main objective was, first and foremost, to provide VR users with the advantages of unrestricted access to their mobile phones in virtual experiences. Therefore, this solution is centered around the potential users, how it could be useful to them, and in what context. For this reason, we based our design and development cycle on the User-Centered Design[1] process, illustrated in

---

[1] User-Centered Design, https://www.usability.gov/what-and-why/user-centered-design.html, (accessed 04/07/2023)

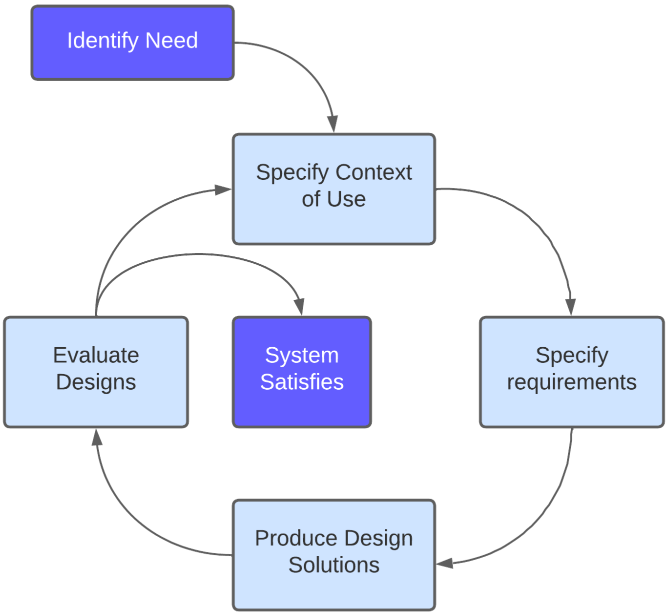


Figure 3.1: User-Centered Design (UCD)

Figure 3.1.

Following the UCD process, the first step is to identify the need. Having identified a need to use smartphones in VR, we then went through the following stages of the process, by defining the context of use and then the requirements of the solution. This allowed us to produce/develop the required solutions and, later on, run the final solution through an evaluation process, to assess whether it successfully satisfies the initial need we identified.

3.1 Context of Use

As we have discussed previously, using external devices during VR sessions is complex and not always easily achievable, almost always requiring a full interruption of the experience. For people that use VR systems and also a mobile phone on a regular basis, the frequent pauses in the experience can be overwhelming, distracting, and distressing, diminishing the potential of both the use of the mobile phone and the VR experience altogether.

Depending on how a person uses their mobile phone, the impossibility of interacting with it inside the VR environment might not significantly impact the experience. An example of this are users who use mobile phones more sporadically and therefore do not need instant and continuous access to notifications. Also, if the tasks that the user needs to perform on the mobile phone are usually very long, tedious or tiresome, it is most likely preferable to momentarily interrupt the VR experience to focus solely on the phone.

Therefore, the contexts of use we had in mind when designing our solution are regular and prolonged VR sessions, in which the VR user wants or needs quick access to their phone at all times to do brief and simple actions. It would be suitable, for example, for users that work inside

VR environments for long periods of time and want to keep up-to-date on communications with other people, whether in the labour or personal context, by quickly checking notifications, reading messages and/or e-mails, and so on. Essentially, focusing on very short tasks for which pausing the VR session would not be justified.

3.2 Requirements

To provide the user with a satisfactory experience of seamlessly using their mobile phone in the virtual environment, they should be able to correctly perform a minimum set of representative tasks that our solution must allow support for, such as:

- See phone notifications
- Open and navigate a messaging application
- Type text messages using the phone's virtual keyboard
- Set alarms and reminders on a calendar application
- Perform searches using an internet browser
- Use the mobile phone in different orientations
- Use the phone's sensors, such as the camera, microphone and fingerprint sensor
- Using other simple applications, such as online shops, news feeds, or social media applications

Having in mind the kind of experience we want our solution to be able to provide and the tasks we aimed to provide support to, we defined its requirements. These requirements serve as a clear set of boundaries that have to be put in place regarding the technical aspects, such as the software and hardware used for the developed solution, as well as the different techniques used. We can organize them by the following different aspects: display, interaction, position, and features.

3.2.1 Display

One of the more relevant parts of smartphones nowadays is their screen. Almost all of the information contained in the phone's system, whether it is user interfaces, media, or even personal data, is provided visually to the user by the screen. Interaction with the phone without access to the screen is, most of the time, impossible. Apart from a few cases, such as when using text-to-speech interfaces or applications for purely auditive or physical interactions, such as phone calls, seeing the phone's display is essential to operate it.

Therefore, the solution must provide a clear visual representation of the content on the phone's screen inside the VR space. Effectively, the user must be able to see, in real-time, a mirrored virtual version of the screen. As we have stated previously in chapter 2, the interactions of the user with the touchscreen will, most of the time, be affected by the responsiveness of the display. For example, when typing text using the keyboard or navigating across different screens, the experience can become frustrating with a significant visual delay in the video stream. Additionally, since much of the information displayed on the mobile phone is usually text, we also have to consider that this virtual screen must provide a relatively high-quality duplication of the original image, allowing for better readability. For these reasons, the solution has to use a video streaming implementation that aims to provide acceptable latency while keeping image quality at a relatively high resolution, ideally the same as the native resolution of the phone's screen.

3.2.2 Interaction

As we discussed in chapter 2, interaction with the mobile device can be achieved through regular means, which is physically operating it, or using indirect approaches, such as inputs in the VR scene or proxy devices. With direct physical interaction, the user can use more of the mobile phone's features, specifically the sensors. For example, if it was impossible to physically handle the device, features such as the camera, fingerprint sensors and even the gyroscope, used for vertical and horizontal orientation of the screen, would possibly be unpractical or very limited. Another factor we considered when setting this requirement was the familiarity that the mobile phone users have with the device. If another device is used to relay the inputs, users will have to learn how to use it and will feel the disconnection between the mobile phone and its virtual version. So, for this work, we opted to define that all the interactions by the user must be done with their hands, directly on the mobile device.

We also discussed that the interactivity of the user with their mobile phone may be impacted significantly by the visibility of the means of interaction, in this case, their hands. To operate the device efficiently, a user needs to have some feedback concerning the position of their hands. This becomes evident when interacting with the touchscreen since, unlike with other interaction methods such as the side buttons or fingerprint sensors, the user has little to no physical clues on where to press the screen. Therefore, a virtual mirror of the position of the hands and fingers may be of significant importance for the solution.

Typing text through the mobile phone's virtual keyboard is a very common task in mobile applications. For this reason, particular attention is given to the precision of the hand and finger tracking. Typing on the keyboard involves pressing possibly many different elements that are often relatively small and close to each other on the screen space. In case the representation of the hands is too disconnected from their reality counterpart, this task might become considerably difficult.

3.2.3 Position

Since the user will be physically handling the device, including the touchscreen and, potentially, other sensors, the device's position in the world is an important factor in connecting the user to the device effectively. This can make the experience feel less artificial and more like its real-world

Name	**Description**
Display visualization	User must be able to clearly see the mobile phone's display rendered in real-time inside the virtual environment.
Physical interaction	User operates the mobile phone by physically interacting with it.
Accurate spacial position	The virtual representation of the mobile phone must be accurate to its real-world counterpart position-wise.
Hand tracking and touch visual feedback	Useful visual feedback, such as the position of the hands and the points of touch, must be available to the user.
Text input accessibility	The virtual feedback on finger/hand position should be precise enough for the user to use the phone's virtual keyboard.
Access to main applications	User must have access to the main functionalities and applications of their mobile phone.

Table 3.1: Requirements

equivalent. Therefore, the device's position must be tracked in real-time as accurately as possible while the user operates it.

Another important factor is that although we must be able to track the position of the phone at all times, the means to do so should not disrupt the experience in a significant way. For example, the tracking device must not get too much in the way of the user's hands while holding the device.

3.2.4 Features

By using the physical device, including interaction with its touchscreen, buttons, and, potentially, its sensors, we can use the device's system integrally at its near full potential, depending on the limitations of the connection to VR. Instead of overriding phone features, such as its applications and system functionalities, with developed alternatives to adapt them to our solution, we aimed to adapt the solution to existing applications and to allow the use of the phone's operating system without changes for the user to perform the list of tasks, mentioned previously in this section.

This requirement is inherently dependent on the other requirements described previously since the use of the phone's real screen output, position, and the ability to interact with it are the factors that enable the user to have access to the usual phone functionalities.

All the main requirements that were described in the previous sections are enumerated and summarized in the table 3.1.

3.3 Development methodology

With the context of use and the initial requirements defined, a design and development cycle was set for the process of experimenting with design solutions and implementing and testing them, following a User-Centered Design process.

Throughout the development, we followed the Agile methodology, dividing the expected work into well-defined phases. Each main component of the solution that had to be implemented was assigned to a different phase or step, namely the points we mentioned before. That is, mirroring the phone's display, tracking its position, tracking the user's hands, and, finally, preparing a testing and data collection stage.

At the end of each phase, we aimed to have a fully functioning new feature, representing a complete iteration of the prototype. To achieve this, we explored different solutions in the first stage, having in mind state-of-the-art technology and available hardware and software. Then, we implemented the chosen solution into the prototype, followed by a testing phase. To test new features, we ran the prototype in the laboratory under normal conditions and observed its behaviour undergoing regular use of the mobile phone. When faced with unsatisfactory results, we turned to another solution and repeated the development cycle.

The methodical development cycle of the proposed solution allowed for more stable control of the quality of the experience in every stage, ensuring every component worked correctly, both on its own and joined with the rest, in an incremental fashion.

3.4 Architecture

To visualize the pretended underlying functioning of the solution, the components that it is composed of, and how they interact and depend on each other, the architecture of the solution is defined and represented in Figure 3.2.

The main components of the infrastructure for this solution are the following:

- **Mobile phone** - The mobile device runs an application responsible for communicating the mobile phone's relevant system information to the VR host application. This includes recording and sending the live video stream of the screen's contents. Additionally, the application can capture the current orientation of the phone, whether it is on portrait or landscape mode, and sends it to the VR application. The application also notifies the VR client of any touch events inside the application, which can be used for touch feedback and calibration of the finger position.

- **Position Tracking Device** - A VR tracking device is physically attached to the phone and, while having a wireless connection established with the host machine, sends real-time data regarding its position coordinates and rotation in the world space, which are obtained for every frame through the VR framework used in the application. Since the device is attached to the phone in a fixed position, the phone's position can be easily calculated through the position of the tracking device.

- **Server** - A server must be running in the host machine or, optionally, on another machine which can establish a network connection with both the mobile phone and the host machine.

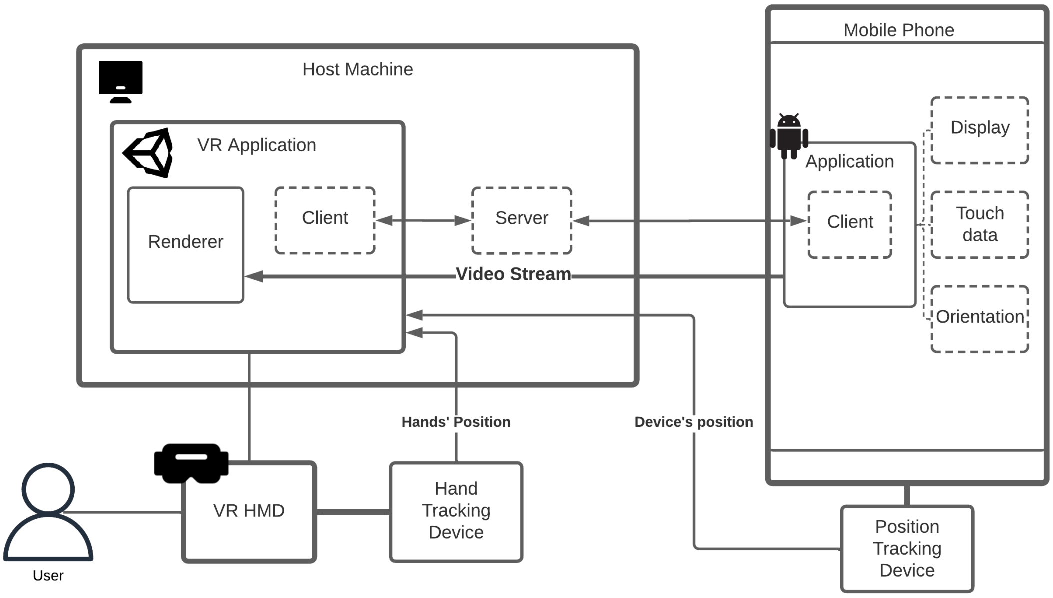


Figure 3.2: Architecture

This server serves as a beacon to both clients, the one in the VR application and the one in the mobile application, through which they can establish the connection between them.

- **Hand Tracking Device** - A hand tracking device containing a depth sensor is attached to the front of the HMD, being able to calculate the position of the hands and fingers of the user in real-time. This data is sent directly to the VR application through a physical connection.
- **VR Application** - The application initiates the connection to the mobile phone through the server and receives all the data mentioned previously, sent by the mobile application and the position tracking device. It uses all the data received to render the phone, including the body and screen, in its correct position according to the tracking device. It is also responsible for rendering the virtual representation of the hands, using the data obtained through the hand-tracking framework.
- **VR HMD** - Interacts directly with the user, displaying the output provided by the host machine's application, including the 3D environment, the phone's virtual representation, and the output of the hand tracking method.

3.5 Summary

In this chapter, we discussed the proposed solution, the mobile phone in VR, and the context and support behind it. We explained the possible contexts of use considered relevant for the problem presented and elaborated deeper on the requirements set for the solution, based on the solutions and experiences we intended to provide. We then presented the development process that we adopted, explaining the methodology used to design, implement and test the solution. Finally, we presented the architecture, including the different components and how they contribute to the final solution.

With a possible solution defined, we proceeded to the realization of a prototype that can be used to evaluate its efficiency in solving the problem, for which the process is described in the next chapter.

Chapter 4

Prototype implementation

Having defined a proposed solution to the problem in chapter 3, including its requirements, architecture and main components, we are able to implement a functioning prototype. With the prototype, we aimed to tackle previously discussed problems and solutions, devising a testable system.

The prototype consists of a system connecting a Unity VR application with a mobile phone through a mobile application and a hosted server, allowing for the inclusion of the phone in the VR environment. When the mobile and Unity applications are started, the mobile phone is represented as a 3D object in the VR scene, including the real-time display of its screen, following its real-world position in the scene. The mobile phone, attached to a tracking device, can be physically operated by the user from within the VR experience, as shown in Figure 4.1, providing most of its usual features. Additionally, a hand-tracking device attached to the HMD tracks the user's dominant hand, and its position and gestures are reflected in the virtual environment in virtual 3D hands. Ultimately, users are provided with a virtual representation of the phone and their means of interaction, illustrated in Figure 4.2.

Throughout this chapter, the process of the prototype implementation is described and organised into sections containing the process for each of the main parts of the solution: the visual representation of the phone, the position tracking, and the hand tracking.

4.1 Visual representation

As discussed previously, the first step into representing the phone inside the VR environment is to have a real-time virtual mirror of the screen. In chapter 2, when researching screen streaming alternatives, we had determined that a real-time communication through the WebRTC protocol was one of the best options regarding image latency. In fact, we have seen that, according to Kyian et al. [14], their application, implementing WebRTC, outperformed all other approaches, including the streaming of screenshots.

Therefore, we implemented an infrastructure to support WebRTC communication between the mobile phone and the VR application, in order to stream the contents of the phone's display. To

Figure 4.1: Final prototype: User operating the mobile phone

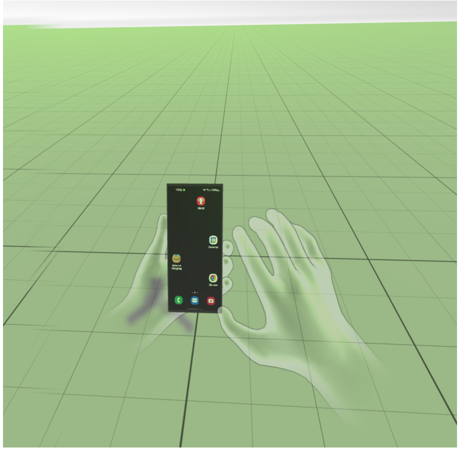


Figure 4.2: Final prototype: VR scene with virtual mobile phone and hands

implement this protocol, we developed three interconnected components: the WebRTC server, the mobile client, and the VR application client.

4.1.1 WebRTC Server

The Web Real-Time Communications (WebRTC) protocol[1] enables peer-to-peer communication, which reduces the latency by removing middleman communication. However, establishing connections between peers requires a server, usually called the signalling server.

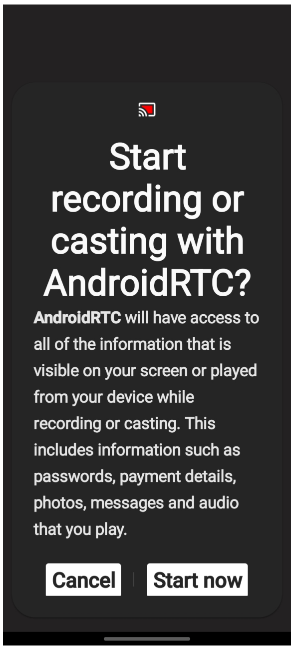
The server is responsible for keeping track of the list of machines that are connected to it and that wish to exchange data through WebRTC communications. When a client wishes to connect to another peer, it makes a request to the signalling server to establish the connection. The server then acts as an intermediary between the two peers, exchanging the messages that initiate the connection, containing the configurations that they must agree upon, such as, for example, the desired video quality, aspect ratio, and framerate.

To implement this server, we based it on an implementation of a WebRTC signalling server, part of the *ProjectRTC - WebRTC Live Streaming*[2] project. The server was developed using *Node.js* and consists of an endpoint that listens for incoming requests using *Web Sockets*. A client may request a list of possible connections, consisting of the clients that have previously requested a connection with the server, and then request a WebRTC connection with the intended client. Additionally, clients can send messages to each other with custom content, apart from the messages strictly related to the execution of the WebRTC protocol.

---

[1] WebRTC, `https://webrtc.org/`, (accessed 04/07/2023)

[2] ProjectRTC, `https://github.com/pchab/ProjectRTC`, (accessed 04/07/2023)



Start recording or casting with AndroidRTC?

AndroidRTC will have access to all of the information that is visible on your screen or played from your device while recording or casting. This includes information such as passwords, payment details, photos, messages and audio that you play.

Figure 4.3: Android application for WebRTC client - Casting permission

4.1.2 Mobile client

We need a client running on the mobile device to connect the mobile phone to the server and, afterwards, to the VR application. This client is responsible for establishing a connection to the signalling server and, therefore, includes the mobile phone as a possible connection for other clients to request. The *ProjectRTC* project also included an Android client, the *AndroidRTC - WebRTC Live Streaming*[3] project, which we also used as the base for our application. Since the last update on the project was about five years ago (September 2018), we had to make many changes to the application due to the design around outdated Android versions. These changes included new privacy and security requirements imposed in newer Android systems regarding the live streaming of the screen. The application was developed in Java and is supported on at least Android 10 to 13 systems.

The mobile application must be compiled with the correct IP address of the signalling server, with which it establishes a connection upon starting it and initiates a media capture service. The address was defined in compile time in the prototype to simplify the interface of the application. The user must give permission to the capture to start every time they start the application, as seen in Figure 4.3. To do this, we use the *MediaProjection* Android token, which grants applications the ability to capture screen contents. The WebRTC protocol implementation was done using a custom WebRTC library for Android, included in the project, that contains the classes and methods required for setting up the connection and configuring the video track with the live capture of the screen.

---

[3] AndroidRTC, `https://github.com/pchab/AndroidRTC`, (accessed 04/07/2023)

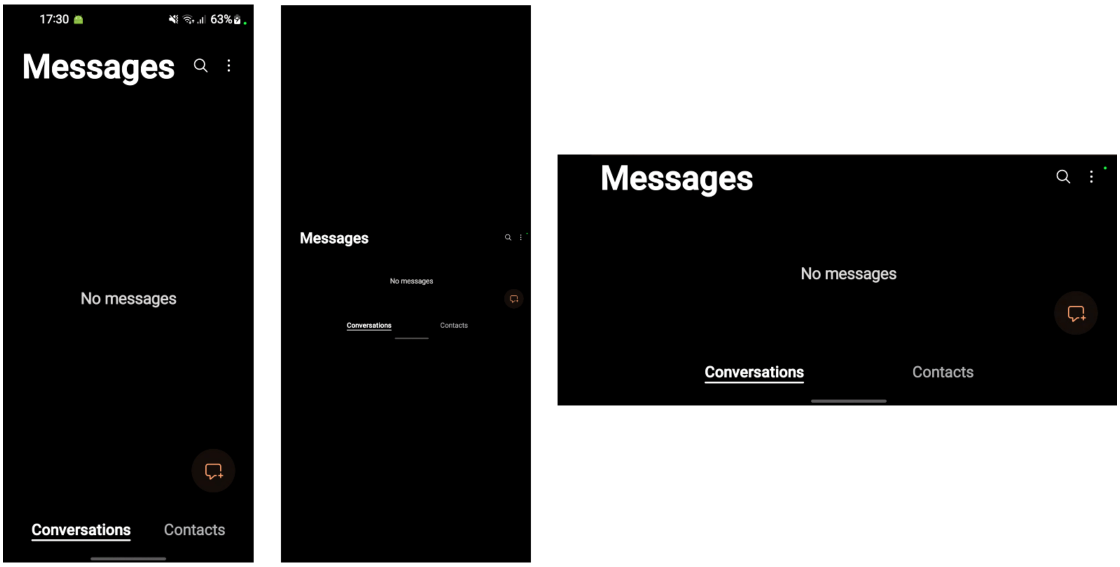
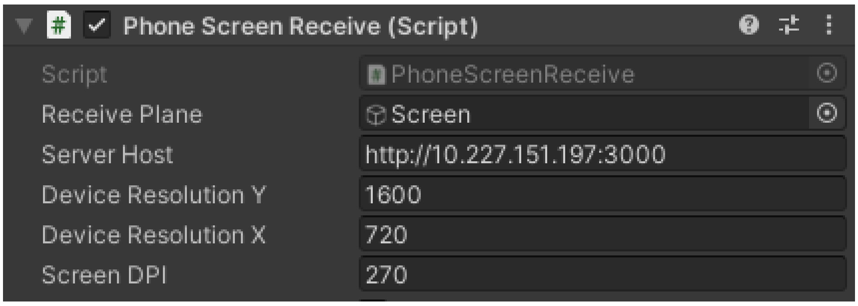


Figure 4.4: Video stream output in different orientations. From left to right: portrait/vertical orientation, landscape/horizontal orientation without fixing the stream's dimensions, and horizontal orientation after fixing the dimensions.

In order to not interfere with the other applications and general interaction with the mobile phone's system and features, the application can be minimized, and the WebRTC client keeps running on a background service. In fact, the application keeps running indefinitely and streaming the screen even when the device is inactive or locked. These factors allow for better usability since the user can use the phone normally and even put it down for extended periods without the need to reset the connection with the VR application every time they unlock the phone. Also, the application is relatively lightweight, containing no graphic elements and only the WebRTC background service, causing only a small impact on the device's performance that is not noticeable by users in most cases.

To mitigate the loss of image quality and, consequently, readability issues with the mirrored image of the screen's contents, the live video stream is configured with the phone screen's native resolution.

Another factor to consider is the change in the phone's orientation. This can be triggered by applications or, most commonly, by physically tilting the mobile phone to a horizontal, also called landscape, orientation. This causes the video stream to have the wrong dimensions since the width and height of the video capture need to be swapped upon orientation changes. If the dimensions are kept, which happens by default, the video content is resized to fit the dimensions of the stream, resulting in a large portion of the video stream output being empty. To fix this, we added to the background service a listener for orientation changes, which notifies the client of the current orientation of the phone so it can modify the dimensions of the video stream. This problem and the solution's output are illustrated in Figure 4.4. Additionally, a message is sent to the VR application to notify it of this change in orientation. This notification is necessary to adjust the phone's VR counterpart properly and will be discussed in the next section.



Phone Screen Receive (Script)

Script	PhoneScreenReceive
Receive Plane	Screen
Server Host	http://10.227.151.197:3000
Device Resolution Y	1600
Device Resolution X	720
Screen DPI	270

Figure 4.5: Phone Screen Receive script - Editor window

4.1.3 Unity application

One of the essential parts of this solution is the VR application that allows for the user's interaction with the mobile phone whilst inside the VR environment. The application is responsible for rendering the live stream of the phone's screen contents into the 3D environment to be visible to the user through the VR HMD. The application was developed using the Unity engine.

To communicate with the WebRTC server and the mobile phone's client, we created a *PhoneScreenReceive* script and added it to the VR scene. The script is responsible for establishing the connection through *Web sockets* when the application initiates. To correctly set the socket connection, we used a wrapper for a *Socket.IO* implementation for .NET clients that works for Unity[4]. Problems with the initial setup of the connection had to be treated with care since the three points, the mobile client, the server and the Unity client must all use the same version of the socket protocol.

Upon connecting to the signalling server through an IP address provided through the Unity editor, the script sends a request to the server to connect to the mobile phone, which must already be connected. From there, the peer-to-peer connection with the phone is made according to the WebRTC protocol through a similar implementation to the mobile application. In this case, we used a WebRTC package[5] available for Unity that contained an implementation of the necessary classes and functions to implement the WebRTC session establishment and receiving of a video stream.

When connected to the mobile phone's client, we obtain a texture containing the live feed of the screencasting, and, to include it in the VR environment, that texture is applied to a 3D plane. In this prototype, the size of the plane can, for simplicity, be edited before compiling the application and must match the exact size of the actual screen. This way, the user can see a plane in the environment that is visually very similar to what the phone's display would look like without the body. The screen plane and other relevant information, such as the server's address and screen dimensions, are provided through the editor, as shown in Figure 4.5.

A simple 3D representation of the device is also added to the environment to enhance the phone's presence in the virtual world, with the screen plane on top of it. The model's size is

---

[4] SocketIOUnity, `https://github.com/itisnajim/SocketIOUnity`, (accessed 04/07/2023)

[5] WebRTC for Unity, `https://docs.unity3d.com/Packages/com.unity.webrtc@3.0/`, (accessed 04/07/2023)

Figure 4.6: Virtual 3D representation of the mobile phone.

adjusted according to the device's real size. The resulting object, shown in Figure 4.6, depicts an entity in 3D space that is recognisable to the users and resembles a smartphone.

Additionally, as mentioned previously, the application is notified if the phone's orientation changes. That is, the video stream dimensions are swapped, and, to keep the correct display of the screen in the 3D plane, the texture must be rotated on the plane according to the orientation. To achieve this, we change the UV mapping of the screen's mesh every time the orientation changes. The side to which the screen is rotated is obtained through the rotation of the phone's object in the VR scene, as its position and rotation are defined according to the position tracking of the real device.

4.2 Position tracking

As we have discussed in the previous chapters, reflecting the real position of the mobile phone in its virtual counterpart impacts the overall interaction of the users with it, as well as the feeling of disconnection between the real and virtual devices. To complement the visual representation of the body and screen of the device in the 3D space, we implemented a way to track the position of the device and use those values in the VR application.

We explored other options through research, such as using traditional VR controllers or fiducial markers. However, these options had some limitations that could decrease the quality of the tracking and of the overall experience, which were explained in chapter 2. To solve this, we opted to try and use smaller tracking devices. For the development of the prototype, we were using HTC

Figure 4.7: HTC Vive Tracker 3.0

Vive equipment and, for the purpose of position tracking, HTC provides a small and light tracking device, the HTC Vive Tracker 3.0[6]. The device is shown in Figure 4.7. In fact, the reduced weight (75 grams) and relatively small dimensions (70.9 × 79.0 × 44.1 millimetres) made it a good candidate for this application.

Support for the tracking device

The Vive Tracker includes IR sensors in many different positions, allowing for precise tracking in many poses and orientations, with a field of view of 240 degrees. To use the HTC Vive equipment, including the headset, a play area must be defined in the room by placing HTC Base stations[7] around the area. These stations emit periodic IR pulses, which, when received by the tracking equipment, are translated into their position in the scene.

The device was not designed to be attached to devices such as smartphones. However, the base, compatible with generic supports such as tripods, allows for the simple development of custom supports that attach the tracking device to various objects. To this end, we designed and assembled our own attachment to the mobile phone, shown in Figure 4.8.

The support is composed of essentially two parts: a grip attached to the phone and a socket for the base of the tracking device. The grip provides a solid attachment of the plastic socket to the phone. However, its most important role in support is to allow the mobile phone's user to hold it comfortably. In this case, we used a *PopSocket*[8] grip. These phone grips are designed specifically for usability and comfort, providing a way for phone users to securely hold their mobile devices with one hand and an easier method for attaching the phone to other objects and surfaces, such as car dashboards. In fact, the phone grip, as its name implies, provides a stronger grip of the user's hand on the device, making it possible to hold it, either vertically or horizontally, without complicated hand positions and with minimal effort. This aspect is very important to this solution

---

[6] HTC Vive Tracker 3.0, `https://www.vive.com/us/accessory/tracker3/`, (accessed 04/07/2023)

[7] HTC Vive Base Station, `https://www.vive.com/eu/accessory/base-station/`, (accessed 04/07/2023)

[8] PopSockets, `https://www.popsockets.com/`, (accessed 04/07/2023)

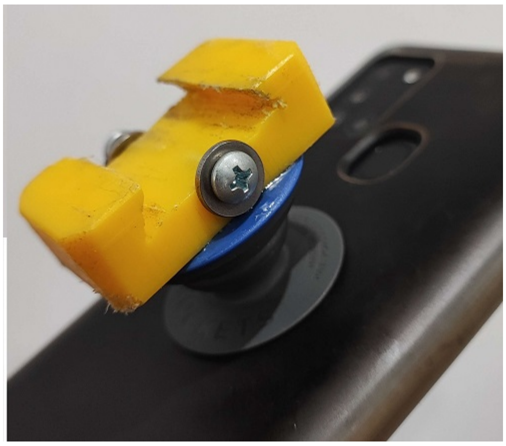


Figure 4.8: Tracker support

since the weight and space added by the tracking device may impact how easy it is to hold the mobile phone.

The use of the grip also means that the contact surface of the support with the back of the mobile phone is relatively small, being just the small diameter of the grip. Beyond making it a less intrusive presence for users, it allows for minimal conflict with elements in the back of the phone.

Glued to the top of the grip, we have a plastic socket with two screws, which we created specifically for the purpose of this support. The socket is sized to fit a generic tripod base, illustrated in Figure 4.9. The tracking device is compatible with the generic base screw, which securely attaches the device to the base. The complete setup of the support with the attached tracking device can be seen in Figure 4.10.

The resulting support does not totally block any features of the phone. In fact, in most smartphones, the user could still use all the physical components and sensors, such as USB ports, fingerprint sensors, headphone jack, IR emitters, and speakers, while having the support attached to the back of the phone. Some sensors, such as the cameras, may be partially blocked depending on the device.

The space between the base of the device and the back of the phone, provided by the grip, also presents a couple of vantages. Firstly, this offset means that the tracker will be some centimetres away from the back of the phone. That means there are fewer positions in which a user could hold the phone that could block the field of view of the IR sensors to the base stations of the play area. During development, we tested directly attaching the device to the back of the phone and holding the phone in a slightly tilted position would make the phone's body significantly block the vision of the tracker to the base stations. Posterior tests using the support we created yielded more satisfying results. Finally, since the user's fingers hold the grip in the middle of that space, the weight of the plastic socket and the tracker is better balanced with the phone's weight.

With the tracking device attached, the implementation in the Unity VR application consisted of using a script provided by the SteamVR plugin for Unity, which is attached to the scenes object that represents the virtual mobile phone. This automatically overrides the position and rotation of

Figure 4.9: Tripod base for tracking device

the 3D object with the tracking data in every frame. By applying the fixed offset corresponding to the distance of the phone to the tracker (length of the support), the virtual phone's position accurately reflects the position of the real phone.

4.3 Hand tracking and visual feedback

We previously explored the idea of enhancing interactivity with the virtual mobile phone by including a visual representation of the user's hands inside the virtual environment. To achieve this, we explored several options.

Initially, we considered some basic implementations of image segmentation algorithms on live camera feeds. The objective was to, using computer vision AI models, filter the video stream from a camera pointed at the user's hands in a way that we could collect only the hands. This way, we could project the image of the hands and, therefore, the user's finger gestures and positions into the virtual environment. To this end, and as hand tracking is not the focus of this work, we experimented with various existing projects that provide real-time image segmentation of skin[9]. Unfortunately, these algorithms proved too heavy performance-wise to be reliable for real-time use on an adequate target framerate and high-resolution footage. Since latency and smooth motion, especially in the context of VR, are critical points for the effectiveness of these techniques in improving the experience, image segmentation had to be discarded. However, it is a technique that could be considered in the future with faster hardware and improved algorithms.

Using software designed especially for hand tracking in VR was then considered. The HTC Vive Pro equipment contains a built-in hand tracking system that uses the front RGB cameras to

---

[9] Deep Learning Techniques for Skin Segmentation on Novel Abdominal Dataset, `https://github.com/MRE-Lab-UMD/abd-skin-segmentation`, (accessed 04/07/2023);
Semantic Segmentation, `https://github.com/WillBrennan/SemanticSegmentation`, (accessed 04/07/2023)

Figure 4.10: Support with the tracking device, attached to the mobile phone

run a computer vision algorithm and, through capturing the hands of the user, build a 3D representation of them. This feature can be accessed through the Vive Hand Tracking SDK[10] and a plugin for Unity is available. However, the first experiments revealed that the output provided by this feature was very unreliable. Probably due to being limited to RGB cameras, the algorithm did not accurately track the hands and would often output unnatural and inaccurate hand poses.

Ultimately, we opted to use hardware dedicated to hand tracking, the UltraLeap Motion device. The device consists of a pair of cameras and IR sensors that, if the user's hands are in its field of view, outputs coordinates in 3D space for the different joints of the hands, including the wrist and finger joints. Figure 4.11 shows an example of the output of the Leap Motion software. The Leap Motion Controller is connected to the machine running the VR application through a USB connection and attached to the VR headset using a mount that is fixed below the front cameras, as seen in Figure 4.12. Due to this device's improved algorithms and hardware compared to the Vive Hand Tracking SDK, its tracking output is much more accurate and precise.

Implementing the hand-tracking software into the Unity VR application was simple since UltraLeap provides a Unity plugin for this effect. A *Service Provider* scripted object was included in the scene, which allows using one of the preset models of hands, properly rigged to work with the joint output given by the Leap Motion device. After preliminary tests using the various representations available, such as solid and skeleton hands, we opted to use transparent hands, as shown in Figure 4.13. These proved to reduce the visual noise and unnecessary distractions while still displaying the overall pose of the hands.

However, the Leap Motion device's tracking is not perfectly accurate. The tracking data becomes fairly inaccurate in certain cases where the hand is not fully visible, which often happens

---

[10] Vive Hand Tracking SDK, `https://hub.vive.com/storage/tracking/overview/`, (accessed 04/07/2023)

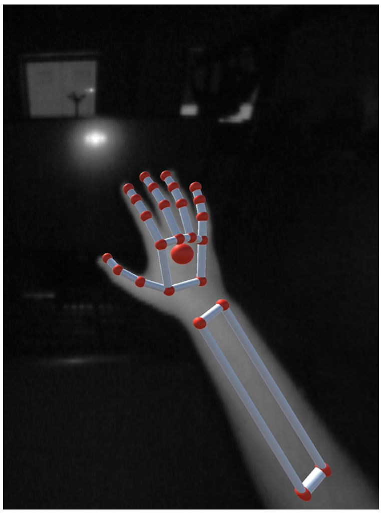


Figure 4.11: Example of Leap Motion Controller output captured in the UltraLeap Console. A 3D representation of the output joints is drawn on top of the Leap Motion Controller's camera feed.

Figure 4.12: VR HMD with attached Leap Motion Controller

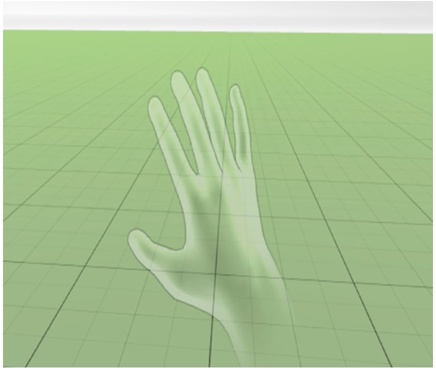


Figure 4.13: Transparent hand in the VR environment

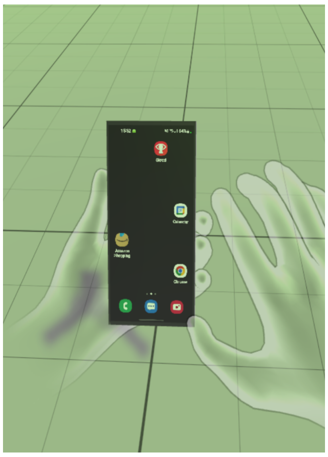


Figure 4.14: Fixed holding hand model (left hand)

with certain hand gestures and usual mobile phone use. For example, if a user holds the phone with both hands and uses their thumbs on the touchscreen, the Leap Motion cameras only have information on the thumbs and, in most cases, that causes the whole tracking to be impossible to obtain. This limits the user only to be able to see their hands when they are not holding the mobile phone and, preferably, are open with their back facing the cameras. To mitigate this limitation, we defined that, for this prototype, users must only use the touchscreen with one hand, preferably their dominant one, and hold the phone with the other.

An option to define whether the user is right-handed or left-handed was added to the VR application. Since the tracking of the hand holding the phone would be very inaccurate, we set the application to display a fixed model of the hand in a holding position, depending on which one of the hands was the user's dominant hand. For example, if the user is right-handed, the left hand would be fixed in the virtual environment holding the virtual phone, as shown in Figure 4.14, while the right hand would be tracked according to the real one.

Calibration process

Another problem was that the position of the hands - the precise 3D coordinates of the joints in the world space - is obtained in relation to the HMD. But the calculation of these positions is almost always not entirely accurate. This is due to slight variations regarding the position offset between the HMD and the Leap Motion Controller. For example, when the device is attached, the position or the tilting angle may vary between sessions. To fix this problem, a calibration process was developed. It is also possible for the position tracking data for the HMD and Vive Tracker to sometimes be imprecise, and even apparently minor differences can significantly impact the precision of the hand.

To adjust the position of the hands and mitigate imprecision, we needed to use a single reference of tracking data in the world space. Since we have the tracking device attached to the mobile phone with an absolute position in the environment, we used its position as the reference. The process consists of the user positioning their hand in a physical spot that we can accurately map to the virtual world based on the reference position of the tracker. This way, we can determine the difference between the position of the virtual hands and the real hands in relation to the reference and apply that difference to the virtual hands to improve their accuracy.

Initially, we attempted to do this by having the user touch, with the tip of their index finger, the corners of the tracker device and use those positions to calibrate. However, this proved too prone to user error since the finger had to be put in a very precise point, and the results sometimes differed significantly. Therefore, we used the mobile phone's touchscreen for the calibration process. The distance from the phone to the tracker is fixed, so from the tracker's position, we can easily get the phone's position, as shown in the diagram in Figure 4.15. We can then use it to get the screen's position and map the physical pixels to virtual positions. So, as an initial stage after starting the application, we need to have the user touch the screen, get the screen position of the touch event, and adjust the position of the virtual hands.

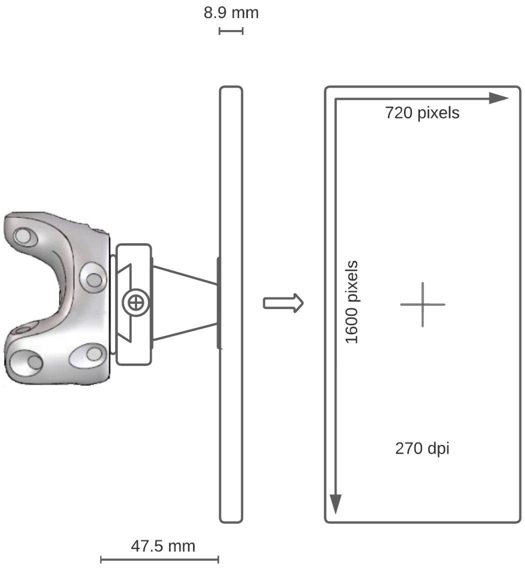


Figure 4.15: Diagram of the positions and measurements of the tracker and mobile phone

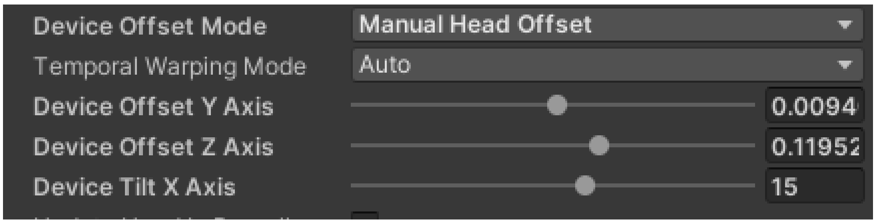
To get the touch data from the mobile phone, we added a touch event listener to the mobile application. Although the application runs in the background for the video streaming service, it is possible to open it, displaying an empty screen. When the application is in the foreground, the user can touch the screen, and the application can capture the touch positions and send them to the VR application through the socket. We considered implementing a calibration process that would run asynchronously at all times, allowing for possibly better accuracy results and would compensate for unpredictable precision problems that could occur during usage. However, the Android system has security policies for capturing touch events, so a running application can only listen to those occurring while it has an activity in the foreground.

Upon receiving a message containing touch coordinates from the mobile application, the VR application stores them, and, in the next frame's update, it calculates the corresponding position on the world space, as shown in the Code Snippet 4.1. It then calls a *Calibrate* method in the *Hand Calibration Screen* script. To adjust the position, we modify the Y and Z offset values in the *Leap Service Provider* script, which the Leap package provides for adjustment and calibration purposes, as also seen in the Editor window in Figure 4.16. As these values are offsets in relation to the axes of the HMD (VR Camera), we calculate the difference between the touch position and the hand's current position and project it onto these axes, as demonstrated on the Code Snippet 4.2.

```
var yDiff = ((-(y - deviceResolutionY/2)) / screenDPI) * INCHES_TO_M;
var xDiff = ((x - deviceResolutionX/2) / screenDPI) * INCHES_TO_M;

var xOffset = xDiff * screenPlane.transform.right;
var yOffset = screenOff.yDiff * screenPlane.transform.up;

```



Device Offset Mode: Manual Head Offset

Temporal Warping Mode: Auto

Device Offset Y Axis: 0.0094

Device Offset Z Axis: 0.11952

Device Tilt X Axis: 15

Figure 4.16: Leap Service Provider - Editor window

```
var screenTouchPoint = screenPlane.transform.position + yOffset + xOffset;
```

Code Snippet 4.1: Calculating world space position from touch coordinates

```
var difference = screenTouchPosition - fingerEnd.transform.position;
var yProject = Vector3.Project(difference, VrCamera.up);
var zProject = Vector3.Project(difference, VrCamera.forward);

var yDifference = yProject.magnitude * (Vector3.Dot(yProject, VrCamera.up) > 0 ? 1 : -1);
var zDifference = yProject.magnitude * (Vector3.Dot(zProject, VrCamera.forward) > 0 ? 1 : -1);

leapXRServiceProvider.deviceOffsetYAxis += yDifference;
leapXRServiceProvider.deviceOffsetZAxis += zDifference;
```

Code Snippet 4.2: Calibration of the hands' position

This calibration solution is less impacted by accidental user error as the reference position we use is obtained directly from actual touch data from the touchscreen and yields better accuracy results upon experimentation. Also, although the calibration process can only be done inside the mobile application, it is possible to open the application at any time and re-calibrate with an indefinite number of touches until the virtual hand's position is sufficiently accurate.

Visual feedback for touch events

To provide additional and even more reliable touch information to the user, we opted to show visual touch feedback on the screen, directly into the mobile device and through the video stream. To this end, we enabled the touch feedback developer option available through the Android system's settings. This displays a white circle centred on the point of the screen that is touched, as shown in Figure 4.17. Therefore, the user can know exactly when and where they have triggered a touch event on the screen. This allows for easy detection of inaccuracy problems and when it might be needed to re-calibrate the hand tracking.

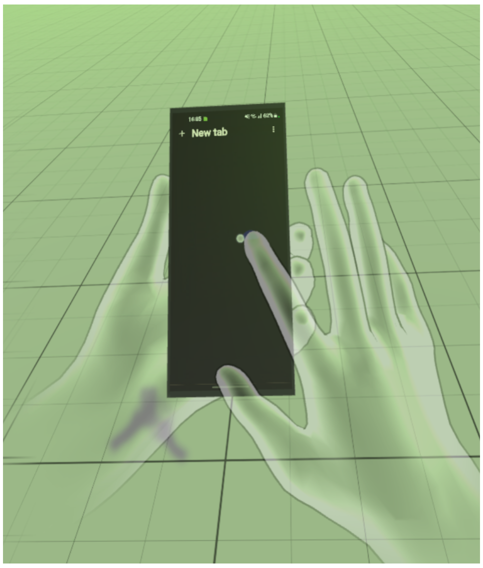


Figure 4.17: Display of visual feedback for touch events

4.4 Summary

In this chapter, we described the final prototype for our proposed solution, including its functioning and features. We went through the implementation process, discussing its various steps, the components integrated into the prototype, and the solutions adopted. The approaches that were taken to solve the problems related to the various aspects of the solution, such as display representation, position tracking, and visual feedback, were discussed and justified, elaborating on the implementation details, limitations, and discarded alternatives. Having completed the final prototype, we proceeded to the evaluation and testing process, as described in the next chapter.

Chapter 5

Evaluation / Experiments

In the previous chapter, we detailed the development of a solution to our problem: effectively using the mobile phone from within a VR environment in a reliable way, minimizing the negative impacts on phone usage performance and providing a similar experience to the regular outside-of-VR use of the mobile device. We aimed to create a system that allows a user to use their physical mobile phone without having to remove the VR headset, using the physical tracking of the different elements - mobile device and hands/fingers - and their virtual representation in the VR environment for improved usability. A set of user-centered tests proved to be essential to properly evaluate the potential of this solution in tackling the problem since the main objective of the solution is to provide a reliable, easy-to-use and accessible experience to its users. The execution of these tests also allowed us to attempt to answer the research questions previously set in chapter 1.

The research questions considered relevant in this work focus on two main points: which mobile phone tasks are possible to do in a VR environment and what techniques we can use to complete these tasks efficiently. Having implemented a few techniques for the solution, we can evaluate the prototype's performance. Apart from technical evaluation of the tools and systems involved, we can get results from user testing to make conclusions on the efficiency of the techniques. To do that, we defined a set of specific tasks that the users must complete with the mobile phone in VR. This list allows for a streamlined way to collect relevant data regarding the execution of certain activities among users. To try and cover most of the interaction possibilities in the mobile phone, the set of tasks was selected based on these different aspects or activities. The broad range of interaction scenarios allows us to better evaluate the feasibility of tasks and the effectiveness of the techniques used.

5.1 Setup

Throughout the tests, a VIVE VR system was used, including an HTC Vive Pro 2 Headset[1] which the participants used to go into the virtual environment. The tests were conducted in a room prop-

---

[1] HTC Vive Pro 2, `https://www.vive.com/eu/product/vive-pro2/overview/`, (accessed 04/07/2023)

erly equipped with all the necessary equipment, including HTC Base stations positioned to form a play area in which the participants could comfortably stand, and a computer for the participants to answer the questionnaires.

Within the play area, precise tracking of the VR devices was possible, including the position trackers, in this case, the HTC Vive Tracker 3.0. The headset contained a Leap Motion attached to the frontal panel, below the cameras, to properly capture the hands of the user. Throughout the tests, the users hold the mobile phone, with the tracker mentioned previously, attached to the support that is placed on the back of the phone.

The mobile phone used for the tests was consistent among all the test instances. For these tests, we used a Samsung Galaxy A21s[2], operating on the Android OS, version 12. Although this possibly impacted the influence that the user's familiarity with the device might have on the tests, the number of variables affecting the results is reduced. Additionally, the support for the position tracking must be attached to the phone prior to testing, and different phones could be incompatible or cause wear on the support. In an attempt to provide a similar experience for all the participants, a brief explanation of the mobile phone's features, navigation and applications was given to the users before starting the tests.

5.2 Tasks

To determine the effectiveness of the implemented techniques for a human-smartphone interaction in VR, the main evaluation process is conducting user tests. For these tests, we set that a user should complete a given set of tasks with the mobile device. These tasks should be representative of common daily actions for phone users and cover a broad set of different types of interactions.

Various interaction requirements and actions allow us to better evaluate the solution's performance under different conditions and better assess the efficiency of the techniques under general everyday smartphone use. Although many different tasks can be done with the device, selecting only a smaller set for the user tests was necessary so that we would not overwhelm the users. Keeping this in mind and considering the important variety of actions mentioned before, we built an extensive list of possible common tasks. We then classified them into different interaction categories, which can be seen in Appendix A.

Among the interaction categories, only one of the tasks was chosen to avoid overextending the final task set while maintaining the variety of actions tested. Another factor to consider while choosing the task set was the dependency of tasks. In fact, some tasks are usually required to execute before others in a normal flow of actions. For example, to read messages on the device, the user must usually open the messaging application beforehand. Additionally, although some tasks contain interactions related to almost all categories, they might be too complex for a user to complete quickly, for example, navigating a map application. They may overwhelm the user, and the restrictions imposed by the VR context, the restricted readability, for example, might make

---

[2] Samsung Galaxy A21s, `https://www.samsung.com/pt/smartphones/galaxy-a/galaxy-a21s-blue-128gb-sm-a217fzbueub/`, (accessed 04/07/2023)

Table 5.1: Tasks for the tests and their categories

Task	**Precise selections**	**Free gestures**	**Anchored gestures**	**Read info**	**Insert info**	**Media manipulation**	**Keyboard**	**Physical manipulation**
Navigate and open applications	Yes	Yes	Yes					
Open contact messages	Yes	Yes		Yes				
Read messages		Yes		Yes				
Use camera	Yes			Yes				Yes
Search for web pages/images through the search engine in the browser	Yes	Yes		Yes	Yes		Yes	
Take a screenshot using power/volume buttons								Yes
Reply to message w/ file/picture	Yes	Yes		Yes	Yes	Yes		
Watch video w/ timeline navigation & horizontal orientation	Yes	Yes	Yes	Yes		Yes		Yes
Search for items in shop application	Yes	Yes		Yes	Yes		Yes	
Add item to cart in shop application	Yes	Yes		Yes	Yes			
Set reminder in calendar	Yes			Yes	Yes		Yes	

the execution of the task difficult and uncomfortable. Finally, we focused on selecting relatively shorter tasks which do not represent a significant interruption in the flow. Tasks that require the user's direct attention for prolonged times may not be justified in this context.

Considering the factors enumerated previously, we set a final list of tasks for the user-based testing, attempting to maximize the categories tested. The list, and the corresponding categories, are enumerated in Table 5.1.

Based on this final list, we created a scenario where users would perform these tasks in a logical order. This way, the users are placed in a realistic situation and more easily understand the flow of the tasks and the actions they must take to progress in the experiment. We defined a scenario in which the user will attend a friend's birthday party, and one of their contacts approaches them to plan everything for the day of the party. The participant is guided by the messages containing implicit instructions on what tasks to do next. The order of the tasks is defined as follows:

- **Task 1** - Open the messaging application to read the message requesting for a photo. Then, take a self-portrait photo using the camera feature built into the messaging application and send it.
- **Task 2** - Go to the internet browser and search for the image of a birthday card. This is done by copying the message containing the search query and pasting it into the browser's search

bar.

- **Task 3** - Take a screenshot of the picture and send it through the messaging application using the gallery feature.
- **Task 4** - Open the video received in a message and navigate to a given time to read an item name that appears on the screen.
- **Task 5** - Go to the Amazon shopping app and search for the item in the video using the search bar and keyboard. Upon selecting one, add it to the cart.
- **Task 6** - Go to the messaging application to read the date of the birthday. Then, go to the calendar application and set a reminder with the friend's name on the right day and month using the keyboard.

5.3 Test Procedure

The experiments consisted of performing a set of tasks divided into two parts with approximately equal estimated times. This allowed us to reduce the probability of overwhelming the participant and increase the quality of the questionnaire answers regarding each task due to the smaller memory span required.

Before starting the tasks, the users signed the consent forms, for both the experiment and the image rights, and the context and objective of the tests were briefly explained to the participants. Then, they were asked to answer a pre-experiment questionnaire. Since we opted to use the same mobile phone for every participant and it was likely that some would not be familiar with that particular device, every user underwent a brief exposition of the system. It was important to ensure that every participant knew the basics for the tests, including the navigation gestures and how to use the physical buttons to take a screenshot. Every application used throughout the test was shown to the user before the test. They were given instructions on what actions they must do to trigger different events, such as selecting a picture from the gallery and a different month in the calendar.

The users were also explained how to hold the device on both orientation modes, depending on their dominant hand. They were asked to use the touchscreen with only one hand's index finger and, to improve hand tracking, keep their hand open, as we had concluded in chapter 4 that this pose had better hand tracking results through preliminary experiments during development.

After the introductory stage, the participants were instructed to place and adjust the HMD and were given the mobile phone. Then the test conductor instructed them to open the calibration application and calibrate the hand, following the procedure explained in chapter 4. The participant could then start performing the tasks, which would be given to them through the messaging app as messages from a contact. The participants performed the experiment while standing, as shown in Figure 5.1, and facing the base stations, so the position tracking of the phone would not be lost.

Figure 5.1: Participant performing the user experiment

Upon completing the third task out of six, the user is asked to fill out a mid-experiment questionnaire with some questions regarding the first set of tasks. Afterwards, they complete the second part of the tasks and answer the post-experiment questionnaire, containing the same questions now focused on the second set, and other questions on the overall experience.

5.4 Data Collection

During the experiments, we collected different data to conduct an analysis of the results of the tests and, therefore, make conclusions regarding the work done on the prototype and support them. We collected both quantitative and qualitative data throughout the tests.

- **Quantitative data** - During the tests, the conductor manually signalled the start and end of each task in the VR application through keyboard input. A script was made to record the time taken to complete each task and, by getting the output from the Leap hand tracking, count the number of frames in which the hand was visible to the user. Additionally, we recorded the live video stream of the phone's screen. Since the touch events were visible in the video due to the visual touch feedback, enabled directly through the operating system's developer settings, we later counted and registered the number of touch mistakes made by every participant in each task.

- **Qualitative data** - Participants were asked to answer three questionnaires before, midway and after the experiment, as mentioned previously. The first questionnaire allowed us to collect the participants' demographic data, including their VR and smartphone experience. The other forms were used to collect data regarding the quality of the experience and the opinion of the users on the different components of the prototype.

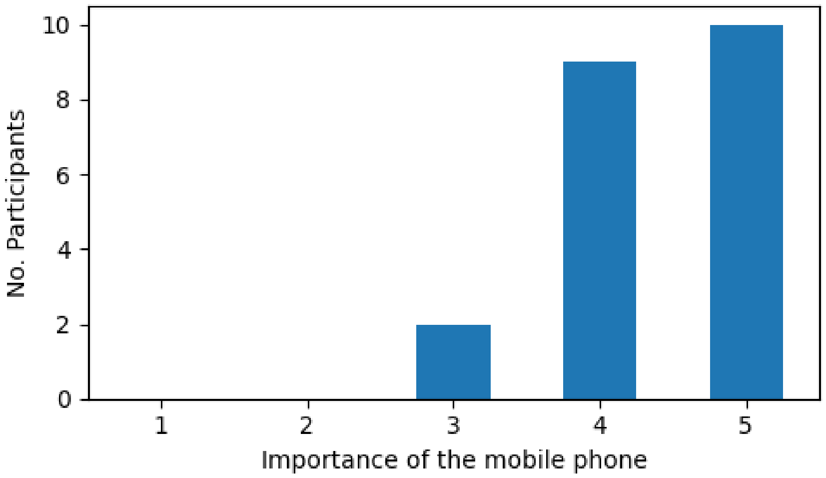


Figure 5.2: Importance of the smartphone in users' lives. The level of importance is displayed on a scale from 1 ("*Insignificant*"), to 5 ("*Very Important*").

5.5 Experiments and Participants

We did a total of 21 experiments with different participants contributing to the user tests, of which 15 are male and 5 are female, aged between 16 and 66 years. Regarding VR experience, only one of the users had had more than 10 VR experiences previously, with also 7 participants having 3 to 10, and 5 of them used it only once or twice. The larger subset (8 users) had never had any VR experience before. This reflects the fact that only one of the users said to have VR equipment at home.

The set of participants, although not having much experience in using VR equipment in general, represent a mostly optimal candidate set for these tests since all participants owned a personal smartphone, and the majority (13 users) said to use it multiple times per hour. The others used it about once an hour (6 users) and the rest (2 users) a few times per day. In fact, 19 users consider the smartphone important or even very important in their lives, as shown in the plot in Figure 5.2.

5.6 Findings

Having collected all the necessary data throughout the user experiments, including objective and subjective data regarding the experience, we performed an analysis of the obtained results, which are presented and discussed in this section.

5.6.1 Touch mistakes

One set of data we collected during the user experiments was the number of errors the users made on each task. For each step of the tasks, we registered the number of touches that represent mistakes, that is, that are not part of the set of touches that are required to progress in the task in that state. For example, in the *Press camera* step, all touches triggered upon the task's start that

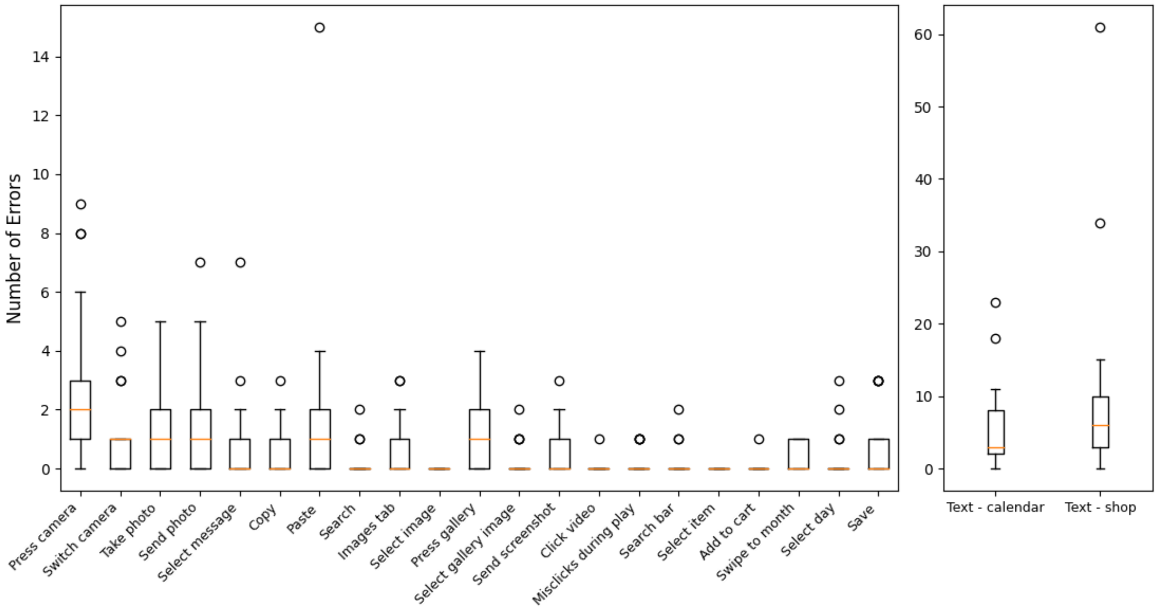


Figure 5.3: Number of errors per task. The tasks are divided into two charts with different scales, due to the much larger amount of errors in the text tasks.

were not on the camera button are counted as mistakes. The number of errors for each sub-task can be seen on the box plot in Figure 5.3.

Not all the tasks required the same number of touches to be completed. For example, the tasks involving the use of the keyboard to write words, the *Text - shop* and *Text - calendar* sub-tasks, required a number of touches equal to the number of letters that had to be inserted. To normalize the error data to compare the mistake ratio among tasks, we defined a baseline, presented in Table 5.2, consisting of the number of touches that would normally be required for the task to be completed. We then obtain the median values of the error-to-baseline ratio for each task, shown in Figure 5.4.

We can see that while most sub-tasks have a null error ratio, which is the desired outcome, some sub-tasks have error ratios up to 200%, meaning a number of errors of about double the touches required to complete it. The first few sub-tasks show a larger error ratio concentration, corresponding to the *Photo* task. A possible justification for this increased ratio of errors is that since these were the initial tasks, the participants were using potentially unfamiliar phones and messaging and camera applications, and were still getting used to the virtual hands and its slight imprecision, they may have been more prone to making mistakes in the first phase. After the first task, and as they get more accustomed to the system, the errors become more dispersed, and the median of the ratios approximates the 100% value.

In fact, we observed during the tests that, right at the start, most users were not expecting any imprecision with the hand tracking and attempted to press the camera button, which is a relatively small element on the screen, very quickly. The confusion due to the lack of feedback on failed touches induced a larger number of mistakes in quick succession. As users became more

Sub-task	**No. Touches**
Press camera	1
Switch camera	1
Take photo	1
Send photo	1
Select message	1
Copy	1
Paste	2
Search	1
Images tab	1
Select image	1
Press gallery	1
Select gallery image	1
Send screenshot	1
Click video	1
Search bar	1
Select item	1
Add to cart	1
Swipe to month	1
Select day	1
Save	1
Text - shop	6
Text - calendar	5

Table 5.2: Baseline for the number of touches per sub-task

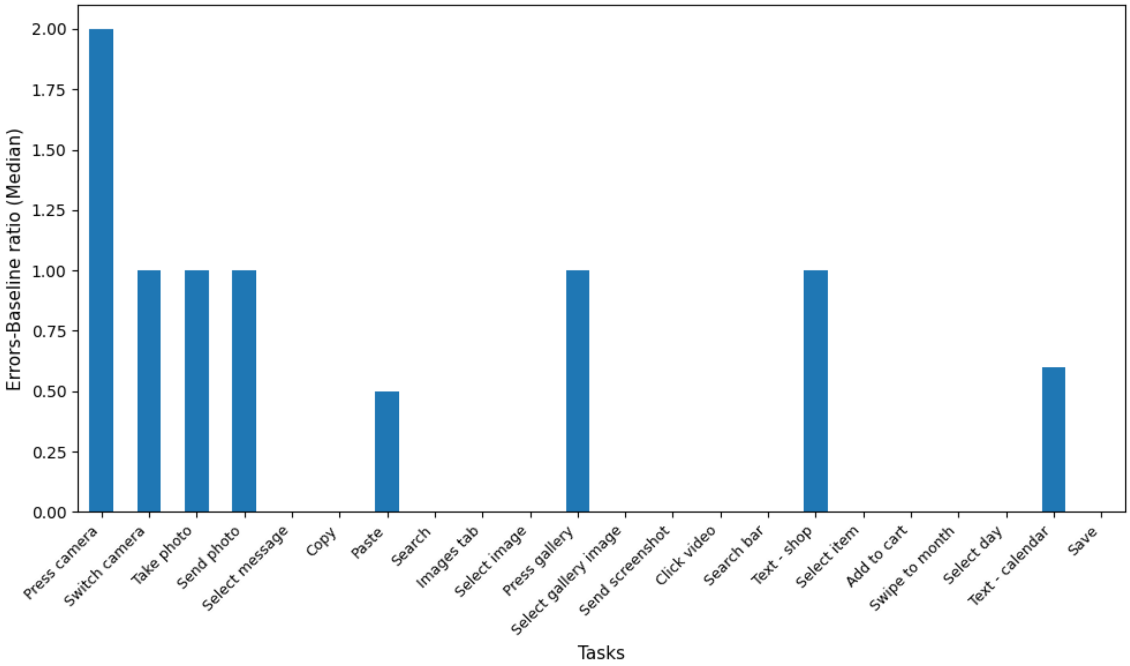


Figure 5.4: Median of the Ratio of Errors to Baseline

familiar with the hands, instances of situations like that became less frequent. Additionally, some of the mistakes in the camera application might have been caused by the input delay. Most of the participants verbally expressed confusion towards this delay, namely when switching the camera and taking a picture. Although this small delay usually happens, even in normal (non-VR) use, this caused mistakes as the users would press the buttons again or try another button under the impression that their input was wrong or didn't register. In fact, although the users had touch feedback on the screen, the buttons in the camera application did not have any immediate visual feedback when pressed, which was most likely the cause for the large number of mistakes in close succession.

Apart from the camera sub-tasks, we can see that the tasks with the largest error ratios are the *Paste* and *Press galery*, as well as the ones that involve typing on the keyboard. These observations could be connected to the touch area of each sub-task, which we explore further in the next subsection.

5.6.2 Touch area size

We noticed during the experiments that the touch actions that participants struggled the most with and, therefore, resulted in greater error ratios, seemed to be the ones that required selecting smaller elements on the screen. To assess the relationship between the size of the elements or buttons that were pressed and the number of mistakes made, we ran a Kendall-tau correlation. It reveals a strong negative correlation between the median of errors and the size of the buttons, which was statistically significant ($\tau = -0.665$, $n = 21$, $p = 0.0002$). This means that the error ratio indeed tends to increase with smaller button sizes. It is also possible to identify this relation in the chart of Figure 5.5, where we have the tasks in crescent order of the average of error ratios and the corresponding button sizes represented in relation to total screen size.

The precision of the hand tracking is the most likely cause of this relation since smaller buttons require more accurate touches.

5.6.3 Task time

We observed large variations regarding the time taken to complete the different tasks. This is demonstrated in the chart in Figure 5.6, which illustrates the Interquartile ranges (IRQ) for each task based on the data collected from the participants' experiments. The times for the *Photo* task seem more consistent, which, although the error ratios were frequently larger in this task, fits with the accustoming and delay factors we previously mentioned regarding these sub-tasks. The errors are more frequent in the beginning but are usually shortly spaced and quickly corrected. In the following tasks, however, the errors seem to cause more unstable execution times, negatively impacting the experience since the tasks should be easy and quick to complete. The relation between the mistakes made by the participants and the time taken to complete the tasks is illustrated in Figure 5.7.

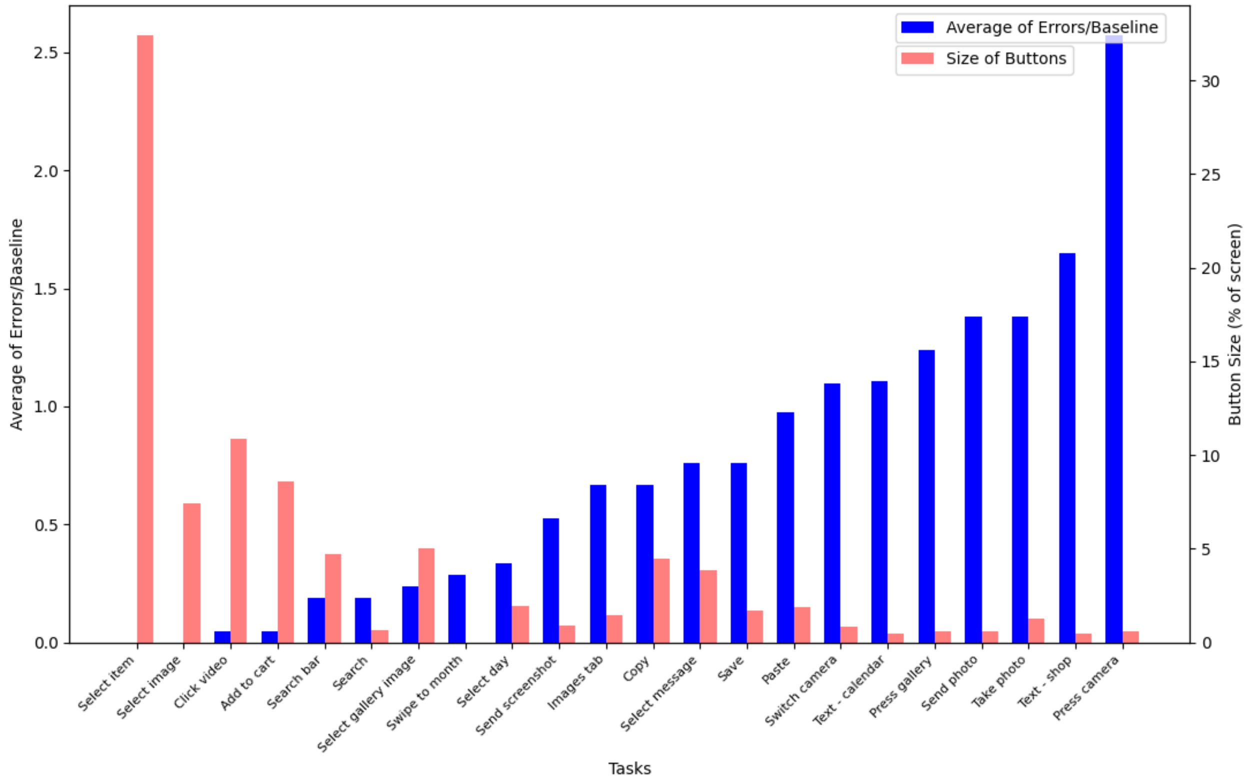


Figure 5.5: Average of errors and button size per task

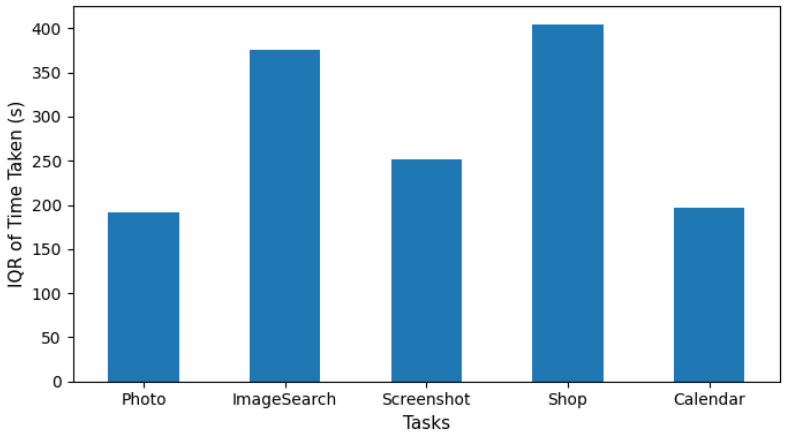


Figure 5.6: Interquartile Range (IQR) of time taken for each task

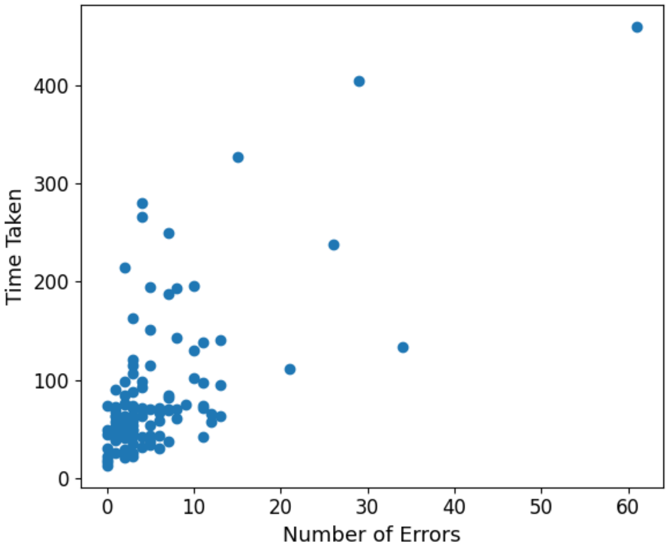


Figure 5.7: Relation between number of errors and time taken

To confirm this relation, a Kendall-tau correlation was run between the number of errors and the time taken per task. The results of this correlation ($\tau = 0.375$, $n = 105$, $p = 4.69 \times 10^{-8}$) indicate that there is a moderate and statistically significant positive correlation between the two variables. This means that when the number of errors increases, so does the time taken, although not in a strictly linear fashion. Considering that the *Photo* task might have been affected by lack of familiarity from the users, a Kendal-tau correlation on only the other tasks was run ($\tau = 0.473$, $n = 84$, $p = 8.86 \times 10^{-10}$), which reflects the factors we discussed, showing a stronger correlation when considering only the other tasks. This shows that, in general, the errors, that, as we have discussed previously, may be induced by the hand-tracking imprecision, cause noticeable impacts on execution times.

5.6.4 Experiment observations and user comments

Although we can make some conclusions based on the objective data collected in the experiments, a large part of the results we observed was the participant feedback. The behaviour of the users while using the prototype in the testing scenario and their opinion on its various components allow for evaluating the quality of the user-centered design of our solution.

Display of the phone

The participants expressed positive feedback on the virtual representation of the phone's screen in the VR environment. We asked the participants the questions "*Did you have any issues reading information on the screen?*" and "*Did the touchscreen feel unresponsive/slow?*", allowing for answers between "*Never*" (1) and "*Very often*" (4). The responses were mostly positive, with a median of 2 and Interquartile ranges of 2 and 0, respectively.

However, some participants expressed concerns about the screen's readability, both verbally and in the open comments of the questionnaire. Since most of the users had not had a VR experience before these experiments, they sometimes lacked the ability to know how to place the headset in the most optimal way to reduce the blur effect caused by the VR lenses, with one user pointing out that "*The largest obstacle was the lack of readability in VR, especially when the head was not perfectly looking at the smartphone*", and another that "*the VR headset might have not been in the best position*". The participants were assisted in placing the HMD correctly and ensuring the image was in focus but throughout the tests the headset can shift slightly and, in some conditions, especially with smaller letters on the screen, the text would become blurry and harder to read. This is, however, a known limitation of the VR device, due to chromatic aberration, halation, field of view, and resolution of the HMD's display [29].

The streaming of the display proved also to be mostly consistent, with only a few cases of a slight delay in the stream, due probably to momentary issues with the internet connection. In fact, only 4 of the participants answered above than 2 to the "*Did the touchscreen feel unresponsive/slow?*" question, and only one user explicitly expressed concern with the delay of the video, saying that "*There was a problem internet connection that made the experience laggy*".

Position of the phone

To assess the performance of the tracking device in obtaining the position of the phone for the virtual environment, we asked participants to answer the question "*Did the position of the virtual phone seem accurate to the real world?*" on a scale of "*Totally innacurate*" (1) to "*Perfectly accurate*" (5). The median of the responses is 4 and the Interquartile range is equal to 0, revealing very positive results in this regard.

Through observing the virtual environment throughout the duration of the experiments, we noticed that there would rarely be a slight shift in the phone's position for a few frames, probably due to the occlusion of the IR sensors of the tracking device in certain positions. However, it did not represent a major impact on the experience based on the answers to the question "*Did the position of the phone randomly shift in an unnatural way?*". On a scale of "*Never*" (1) to "*Very often*" (5), the median of responses was 2 with an Interquartile range of 2.

Holding the phone and physical buttons

For the position tracking, we had to possibly compromise some aspects of the comfort while holding the phone. We observed that the participants quickly learned how to hold the device with the support for the tracking device attached to the back, although sometimes it was a bit confusing for the user to hold it at first when it was handed to them. To evaluate the experience of the participants with this device, we set some questions on it. The questions and the corresponding results are shown in Table 5.3.

We can see that the results for the user's comfort were mostly positive. The participants did not consider the extra support attached to the phone to be too intrusive to the experience. Also, during the experiments, it was noticeable that they quickly got accustomed to it once instructed on

Question	**Scale**	**Median**	**IRQ**
How comfortable did you feel holding the phone? [Portrait / Vertical orientation]	"*Very uncomfortable*" (1) to "*Very comfortable*" (5)	4	1
How comfortable did you feel holding the phone? [Landscape / Horizontal orientation]	"*Very uncomfortable*" (1) to "*Very comfortable*" (5)	4	2
How did the weight of the phone feel?	"*Lighter than usual*" (1) to "*Heavier than usual*" (5)	3	1
How bothered or restrained did you feel by the device attached on the back of the phone?	"*Not bothered*" (1) to "*Very bothered*" (5)	2	2

Table 5.3: Answers on holding the phone, including the scale used for each answer, the median and the Interquartile range.

how to hold the phone. Regarding the *Screenshot* task, users were asked to answer their opinion on the statement "*The devices attached to the phone did not interfere with the task*" using the Likert scale. The median of responses was "*Strongly agree*" (5) with an Interquartile range of 2.

However, the total weight of the device with the tracking device is a problem expressed by the participants, with a more mixed result in the questionnaire. One of the comments left by the participants was that "*Holding the phone for a long time was hard because it strained the muscles. Not because of the thing attached, but because it was heavy and had to be in more or less the same position and held vertically*". Although the tracking device by itself is not very heavy, the difference is apparent to the users, and it can be tiresome to hold the phone for extended periods of time. Additionally, one of the users pointed out that, although they felt comfortable enough holding the phone, switching between the orientation modes was not optimal because of the extra weight and devices.

Regarding the interaction with the physical buttons on the side of the phone, the participants generally did not demonstrate much effort to press the two buttons to take the screenshot. Since the users can feel the buttons with their fingers, it is easier to press them without any visual cues in the virtual environment. To the question "*The interaction with the physical buttons felt normal*", using the Likert scale, the median of responses was "*Agree*" (4) with an Interquartile range of 1. The users only resorted to more visual cues to other actions in which there was not a presence of any physical cues, such as using the touchscreen.

Hand tracking and interaction

The hand-tracking component of the solution is the one for which the participants openly expressed more questions, doubts, and problems, as it was the only visual connection they had inside the virtual environment to their means of interaction with the phone.

During the experiments, we could observe that mistakes related to misunderstandings of the flow of the tasks were rare and, instead, mostly caused by small imprecisions in the tracking of the finger used to operate the touchscreen. Participants would often be confused, especially in the first tasks of the experiment, asking questions such as "I'm pressing the button, but it's not doing anything" or "I put my finger on the camera, but the phone opened the gallery selection tab".

The hand-tracking system displayed inconsistent precision depending on the pose of the hands, so when the users unconsciously changed it, the virtual finger would be significantly misaligned with the actual touch display. This is reflected in the question "*How difficult was it to use the touchscreen, compared to normal out-of-VR use?*". Using a scale from "*Much easier than usual*" (1) to "*Much harder than usual*" (5), the median response was 4, with an interquartile range of 0. When faced with the statement "*The precision of the fingers was satisfactory enough to use the screen*", 9 out of the 21 participants answered either "*Strongly disagree*" (1) or "*Disagree*" (2), with a response median of "*Neutral*" (3) and Interquartile range of 2. The limitation on the hand position and pose imposed by the Leap Motion device to be able to track the gestures correctly also transpired as a detriment to the experience to some users, with a participant stating "*the way of using the touchscreen felt unnatural as I usually use my thumbs and not my index finger*".

The imprecision of the hand-tracking solution caused certain tasks to be more challenging. Participants consistently struggled and expressed frustration with the keyboard typing tasks due to having to press small elements on the screen in a given order, with tight spaces between the different elements. We have shown that the median number of errors per required selection (error-to-baseline ratio) is not high. Still, the user feedback shows that a slight imprecision can cause significant detriment to the experience, increasing frustration and, in some cases, the amount of time need to type a short word correctly. A participant stated, "*if the calibration is slightly off, typing on the keyboard becomes a complex task*". So we have various factors pointing out that this method is not ideal for selecting small elements on the screen with consistent precision.

Nevertheless, participants agreed that the virtual representation of the hands did contribute positively to the experience, with a median response of "*Agree*" (4) to the sentence "*The virtual hands helped to get a stronger connection with the phone and reality*" (IRQ = 1, Likert scale). Additionally, the majority of users (12 out of 21) considered the virtual hands to be essential for the experience, with a median response of "*Agree*" (4) to the sentence "*The virtual hands were essential for the use of the phone in VR*" (IRQ = 2, Likert scale).

The display of tactile feedback through the white indicator on-screen also proved to be indispensable in the experiments. With the imprecision of the virtual fingers, the touch indicator was sometimes the only visual cue for the participants to know where to touch, especially for smaller elements on the screen. They very often used the indicator to have some notion of how far apart the virtual and real fingers were, and one participant stated "*it allowed me to determine how much I had to compensate to a certain side when typing*". A learning factor is, therefore, in action, which was even expressed by the users, that sometimes stated that once they got used to the lack of precision it became easier to select the smaller elements. We can see that on the reduced error ratio on the calendar text task when compared to the previous text typing task in the shop application, illustrated in Figure 5.4.

Overall experience

The participants were asked, at the end of the experiment, whether they would want to have access to our solution if they used VR equipment frequently. The majority (15 users) answered affirmatively, with 2 users replying negatively and the rest (4 users) being unsure. Some of the participants stated that this solution would be useful for quick tasks on the phone, avoiding taking off the VR equipment regularly. The most negative feedback was regarding the keyboard-typing tasks, which some users stated were genuinely difficult due to tracking imprecision and the worst parts of the experiment.

5.7 Summary

In this chapter, we defined the evaluation process for our solution, consisting of user-centered experiments, and explained their setup. We set the list of different tasks that comprised the duration of the experiment, based on various parameters, in order to maximize the phone use scenarios included. With the tasks defined, we elaborated on the testing procedure and the process of data collection.

With the data obtained from the experiments, we analyzed the different aspects of the experience, such as the mistakes made during tasks and the time taken, and dove deeper into the feedback from the participants. We concluded that the majority of the participants had an overall satisfactory experience, completing the tasks without many errors, with the majority (15 out of 21) of them stating that they would use this in future VR experiences. Certain tasks, however, are not easily performed and some adjustments to the system are necessary, namely the hand-tracking component.

Chapter 6

Conclusions and Future Work

Virtual Reality provides an unprecedented means to immerse its users into an entirely virtual environment and, therefore, a set of possible interactions and scenes that can be easily accessed. VR equipment, such as headsets, is designed to create a barrier from the real world, as reducing sensorial input from our physical surroundings helps increase immersion in the virtual world. However, users might want to perform actions outside of the VR experience, such as using their mobile phone, without having to leave it. This concept of including real elements in the virtual world is called Augmented Virtuality (AV), which was the focus of this work.

As the mobile phone is present in most people's and, therefore, most VR users' everyday lives nowadays, we opted to focus on bringing this type of device into VR environments. We researched the main topics around mobile phone uses in VR, analysing previous related works. We focused our research on interaction in VR and integrating external devices into virtual experiences and collected the techniques already developed for this purpose. We explored various topics, such as the mirroring of the phone's display into VR scenes and tracking of the position of the device to map it to virtual spaces. Additionally, we explored other AV techniques, such as including the user's fingers in the virtual environment along with the mobile phone.

From the research made and in response to the problem, we proposed to develop a system that integrates the mobile phone into VR environments. Its main objective was to allow VR users to interact with a virtual version of their mobile phones without breaking the VR experience. We then defined an architecture for the solution based on the techniques we gathered in the research phase and developed a prototype. The final prototype provides VR users with a system that connects a mobile phone to a VR application, displaying an approximate real-time virtual representation of the phone inside the VR environment.

To evaluate the suitability of the techniques implemented and the quality of the phone usage experience in VR, we performed user-centered evaluation experiments on the prototype, where participants were asked to perform a set of tasks, representative of common uses of the smartphone. Having analysed the results of the tests, we concluded that the developed prototype provided a solid integration of the phone in VR, yielding an overall acceptable error ratio for short-duration phone tasks. While there are some limitations on the equipment used, such as minor tracking imprecision and limited text readability through the VR headset, we managed to achieve our objective of reliably integrating the mobile phone, and its essential features, into virtual environments.

Future work

The final prototype developed for this work successfully satisfies the requirements we had previously set and integrated all the components we had planned. Nevertheless, after the development and evaluation, we noted some changes and improvements that can be explored in the future.

The applications created for the prototype, namely the mobile application and the VR application, lack some features that are useful for a regular VR user that would use this solution. As we have mentioned before, the applications must be compiled beforehand with a set of data, such as the server's IP address and the mobile phone's size. A user interface can be added to the applications to insert a custom address and, for the mobile application, a name or identifier field could possibly be inserted by the user. This way, if the user has multiple devices, they could identify them through the list of connected devices provided by the server and, in the VR application, select the one they want to connect to. The size of the mobile phone's body could also be inserted through the user interface, for a more customised experience, easily allowing for a greater variety of devices.

Also, one of the problems that emerged during the user experiments was the poor readability of text on the phone's screen under certain circumstances. If the HMD was not placed correctly on the user's head or if the text was of a smaller font, the participant would struggle to try to read it. This may have been caused, in part, by the VR headset's limitations, as we have discussed in Chapter 5. Although we tried to mitigate it by changing the text font to the larger size available in the operating system's settings, some instances of text in the applications were still unreadable to some users. To further improve the experience, it would be possible to enable more accessibility options, such as zooming on parts of the screen, which are present in Android systems, and ensure that the users know how to use them when necessary. We also discussed implementing other viewing methods in the VR space, such as toggling a larger virtual screen in the virtual environment for better reading.

Finally, the most prominent problem identified throughout the user experiments was the difficulty that some users had while interacting with the touchscreen, mostly related to the hand-tracking component. Even after a calibration process, the Leap Motion hand-tracking module's output still presented some minor imprecision, which, for inputs that require more precision, such as text-typing, significantly impacts the experience. In the future, other options to make the hand-tracking process more robust may be implemented. An automatic periodic calibration process could be added using, for example, image recognition methods on the screen video stream to identify touch events and re-calibrate based on the coordinates. Image recognition algorithms can also be explored in identifying the screen and the finger's positions in relation to it. Although the image recognition projects we experimented with during development did not yield satisfactory results in real-time, it is an approach to consider in future projects. Additionally, we have previously concluded that there appeared to be an adapting/learning process for the users with the virtual hands, yielding better results after the first few tasks as participants grew accustomed to the eventual imprecision. Therefore, a hand-tracking familiarization phase before testing could prove useful in future works.

The suggested improvements we enumerated could help create a system that allows users to see and operate the mobile phone from within virtual environments, providing an accessible, customised and robust experience.

References

[1] Luis Afonso, Paulo Dias, Carlos Ferreira, and Beatriz Sousa Santos. Effect of hand-avatar in a selection task using a tablet as input device in an immersive virtual environment. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 247–248, 2017.

[2] Ghassem Alaee, Amit Deasi, Lourdes Peña-Castillo, Edward Brown, and Oscar Meruvia-Pastor. A user study on augmented virtuality using depth sensing cameras for near-range awareness in immersive VR. In *IEEE VR's 4th Workshop on Everyday Virtual Reality*, 03 2018.

[3] Tatsuya Amano, Shugo Kajita, Hirozumi Yamaguchi, Teruo Higashino, and Mineo Takai. Smartphone applications testbed using virtual reality. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '18, page 422–431, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Vladislav Angelov, Emiliyan Petkov, Georgi Shipkovenski, and Teodor Kalushkov. Modern virtual reality headsets. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5, 2020.

[5] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. Bringing full-featured mobile phone interaction into virtual reality. *Computers and Graphics (Pergamon)*, 97:42–53, 6 2021.

[6] Amal Benzina, Marcus Toennis, Gudrun Klinker, and Mohamed Ashry. Phone-based motion control in VR: Analysis of degrees of freedom. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, page 1519–1524, New York, NY, USA, 2011. Association for Computing Machinery.

[7] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers. *IEEE Transactions on Visualization: Computer Graphics*, 26(12):3490–3502, 12 2020.

[8] Michael Chan. Mobile phones and the good life: Examining the relationships among mobile use, social capital and subjective well-being. *New Media and Society*, 17:96–113, 1 2015.

[9] Joon Chuah and Benjamin Lok. Experiences in using a smartphone as a virtual reality interaction device. *International Journal of Virtual Reality*, 11:25–31, 01 2012.

[10] Amit P. Desai, Lourdes Peña-Castillo, and Oscar Meruvia-Pastor. A window to your smartphone: Exploring interaction and communication in immersive VR with augmented virtuality. In *2017 14th Conference on Computer and Robot Vision (CRV)*, pages 217–224, 2017.

[11] Yuan Du, Haoyi Ren, Gang Pan, and Shjian Li. Tilt & touch: Mobile phone for 3d interaction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, page 485–486, New York, NY, USA, 2011. Association for Computing Machinery.

[12] Youngwon Kim and Gerard Kim. HoVR-Type: Smartphone as a typing interface in vr using hovering. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 200–203, 01 2017.

[13] Alena Kovarova and Maros Urbancok. Can virtual reality be better controlled by a smart phone than by a mouse and a keyboard? In *Proceedings of the 15th International Conference on Computer Systems and Technologies*, CompSysTech '14, page 317–324, New York, NY, USA, 2014. Association for Computing Machinery.

[14] Stanislav Kyian and Robert Teather. Selection performance using a smartphone in VR with redirected input. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction*, SUI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[15] Yang LI, Jin HUANG, Feng TIAN, Hong-An WANG, and Guo-Zhong DAI. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019.

[16] Nicholas G. Lipari and Christoph W. Borst. Handymenu: Integrating menu selection into a multifunction smartphone-based VR controller. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 129–132, 2015.

[17] Xiaoxu Liu, Xiaoyi Feng, Shijie Pan, Jinye Peng, and Xuan Zhao. Skeleton tracking based on kinect camera and the application in virtual reality system. In *Proceedings of the 4th International Conference on Virtual Reality*, ICVR 2018, page 21–25, New York, NY, USA, 2018. Association for Computing Machinery.

[18] Akhmajon Makhsadov, Donald Degraen, André Zenner, Felix Kosmalla, Kamila Mushkina, and Antonio Krüger. VRySmart: A framework for embedding smart devices in virtual reality. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery.

[19] Fabrice Matulic, Aditya Ganeshan, Hiroshi Fujiwara, and Daniel Vogel. Phonetroller: Visual representations of fingers for precise touch input with mobile phones in VR. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–13, 05 2021.

[20] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. A dose of reality: Overcoming usability challenges in VR head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2143–2152, New York, NY, USA, 2015. Association for Computing Machinery.

[21] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In Hari Das, editor, *Telemanipulator and Telepresence Technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, SPIE, 1995.

[22] David Nahon, Geoffrey Subileau, and Benjamin Capel. "Never Blind VR" enhancing the virtual reality headset experience with augmented virtuality. In *2015 IEEE Virtual Reality (VR)*, pages 347–348, 2015.

[23] Thammathip Piumsomboon, Gun Lee, Robert W. Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39, 2017.

[24] Muhammad Sarwar and Tariq Soomro. Impact of smartphone's on society. *European Journal of Scientific Research*, 98, 02 2013.

[25] Gian-Luca Savino. Virtual smartphone: High fidelity interaction with proxy objects in virtual reality. *CoRR*, abs/2010.00942, 2020.

[26] Jongkyu Shin and Kyogu Lee. Incorporating real-world object into virtual reality: using mobile device input with augmented virtuality. *Multimedia Tools and Applications*, 8 2022.

[27] Jeongmin Son, Sunggeun Ahn, Sunbum Kim, and Geehyuk Lee. Effect of contact points feedback on two-thumb touch typing in virtual reality. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery.

[28] Tomomi Takashina, Mika Ikeya, Tsutomu Tamura, Makoto Nakazumi, Tatsushi Nomura, and Yuji Kokumai. Real-virtual bridge: Operating real smartphones from the virtual world. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, page 449–452, New York, NY, USA, 2018. Association for Computing Machinery.

[29] Niteesh Yadav. Technical challenges for typography in AR/VR. `https://fonts.google.com/knowledge/using_type_in_ar_and_vr/technical_challenges_for_typography_in_ar_vr`. Accessed: 04-07-2023.

[30] Li Zhang, Huidong Bai, Mark Billinghurst, and Weiping He. Is this my phone? Operating a physical smartphone in Virtual Reality. In *SIGGRAPH Asia 2020 XR*, SA '20, New York, NY, USA, 2020. Association for Computing Machinery.

[31] Yuxuan Zhang, Hexu Liu, Shih-Chung Kang, and Mohamed Al-Hussein. Virtual reality applications for the built environment: Research trends and opportunities. *Automation in Construction*, 118:103311, 2020.

Appendix A

Possible tasks for experiments

Task	**Precise selections**	**Free gestures**	**Anchored gestures**	**Read info**	**Insert info**	**Media manipulation**	**Keyboard**	**Physical manipulation**
Open notification tray			Yes					
Read notification				Yes				
Navigate and open applications	Yes	Yes	Yes					
View and close open applications		Yes	Yes					
Open contact messages	Yes	Yes		Yes				
Read messages		Yes		Yes				
Reply to message	Yes			Yes	Yes		Yes	
Reply to message w/ file/picture	Yes	Yes		Yes	Yes	Yes		
Use camera	Yes			Yes				Yes
Phone call	Yes	Yes		Yes				
Watch video w/ basic controls		Yes		Yes		Yes		
Watch video w/ timeline navigation	Yes	Yes	Yes	Yes		Yes		
Watch video w/ horizontal orientation	Yes	Yes		Yes		Yes		Yes
Watch video w/ timeline navigation & horizontal orientation	Yes	Yes	Yes	Yes		Yes		Yes
Open file/picture	Yes	Yes						
Pan/Zoom on picture	Yes	Yes		Yes		Yes		
Navigate through web-page on browser	Yes	Yes		Yes				
Search for web pages/images through the search engine in the browser	Yes	Yes		Yes	Yes		Yes	
Navigate social media feed		Yes		Yes				
Navigate news feed		Yes		Yes				
Write a social media post	Yes			Yes	Yes		Yes	
Set reminder in calendar	Yes			Yes	Yes		Yes	
Search route in maps	Yes	Yes		Yes	Yes	Yes	Yes	
Search for items in shop application	Yes	Yes		Yes	Yes		Yes	
Add item to cart in shop application	Yes	Yes		Yes	Yes			
Unlock phone with power button								Yes
Adjust system volume								Yes
Take a screenshot using power/volume buttons								Yes

Table A.1: Possible tasks and categories

Appendix B

Test results

B.1 Time taken for tasks

No. Test	Photo	ImageSearch	Screenshot	Video	Shop	Calendar
1	37.70436478	28.23332977	21.26246834	71.93770599	133.5344696	57.69470215
2	63.7386322	404.1578674	66.76790619	56.73023987	459.2190552	82.21715546
3	33.89056015	83.97668457	74.47039795	73.78141785	137.8986969	49.11879349
4	44.02392197	30.15407181	63.03769004	130.4701843	72.74048662	69.33053374
5	114.8131714	151.6316681	57.59891701	60.40764236	63.82234931	43.06535339
6	22.08088684	49.61665726	22.88240623	90.47046661	70.35845184	73.89013672
7	75.43921661	59.29090881	18.88353729	50.08914185	102.1286774	63.22866821
8	65.47981262	70.51873779	44.80725861	46.99858093	71.09816742	59.28879929
9	37.68970108	120.7851868	63.98641205	68.78153229	84.73983765	71.83678436
10	42.51140976	40.05759811	31.60342407	52.10632324	69.95896149	111.5624084
11	42.74233246	46.24869919	24.41455841	39.04561996	54.11770248	41.19635773
12	130.5496674	194.1212006	71.72016144	106.1498184	187.2952118	87.5737381
13	141.2348022	280.1830139	265.6218872	149.9051514	107.1931915	195.7226105
14	62.93562698	114.4211578	39.35567474	37.24580765	74.22937775	69.74011993
15	30.91636658	56.79965973	30.07417107	93.72915649	142.9357452	48.10852432
16	213.9862061	163.1270447	53.20838547	59.16043854	327.3767395	237.8402252
17	94.54680634	193.2759247	76.4858551	115.1432571	249.544281	90.5402298
18	35.83335495	98.1251297	26.45851517	66.85767365	71.80016327	49.29550171
19	42.5782814	92.93517303	26.09919167	41.12208557	97.02412415	61.20597458
20	24.60351563	53.28194809	13.49616718	86.76550293	75.27589417	44.64334869
21	43.43569946	68.52313995	16.97310066	62.88843775	99.09133148	53.85155869

Table B.1: Time taken for tasks

B.2 Mistakes in sub-tasks

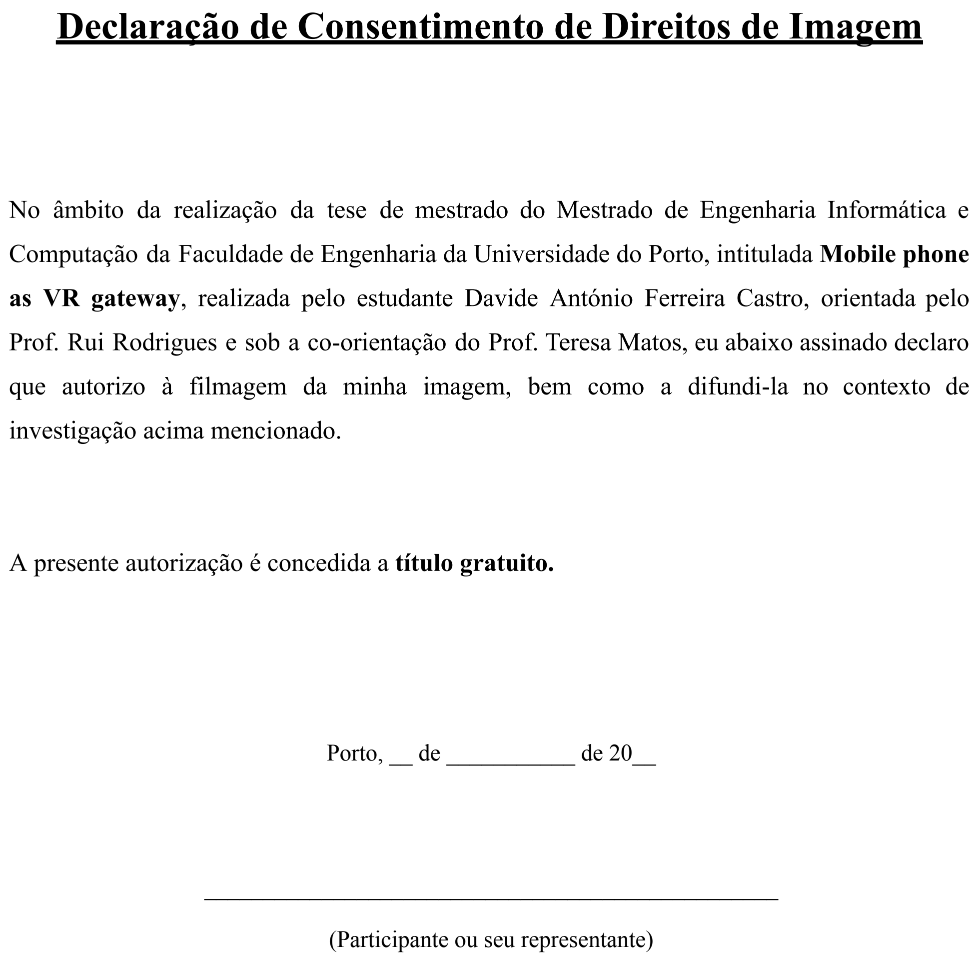
No. Test	Press camera	Switch camera	Take photo	Send photo	Select message	Copy	Paste	Search	Images tab	Select image	Press gallery	Select gallery image	Send screenshot	Click video	Misclicks during play	Search bar	Text - shop	Select item	Add to cart	Swipe to month	Select day	Text - calendar	Save
1	2	0	1	2	0	0	1	1	0	0	1	0	1	0	0	0	34	0	0	0	0	9	3
2	3	0	0	1	7	3	15	1	3	0	3	1	2	0	0	0	61	0	0	1	0	5	1
3	1	0	1	3	1	0	1	0	0	0	1	2	0	0	0	0	11	0	0	0	0	3	0
4	0	1	2	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	3	1
5	0	3	0	0	0	1	3	0	1	0	1	0	0	0	1	0	3	0	0	0	3	3	0
6	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	11	0
7	6	1	1	1	3	0	1	0	2	0	0	0	0	1	0	0	10	0	0	1	1	0	0
8	1	4	0	7	0	1	4	0	0	0	2	0	1	0	0	0	3	0	0	0	0	3	0
9	2	1	4	0	1	1	0	0	1	0	1	0	0	0	0	0	7	0	0	1	0	10	0
10	0	1	1	0	0	0	1	0	0	0	4	0	0	0	0	0	8	0	0	0	2	18	1
11	2	1	0	1	0	0	1	0	0	0	1	0	1	0	0	0	5	0	0	0	0	2	0
12	8	1	0	1	0	1	4	0	0	0	0	1	3	0	0	1	6	0	0	1	1	1	0
13	9	0	1	3	2	1	1	0	0	0	3	1	0	0	1	0	3	0	0	1	0	6	3
14	8	0	5	0	1	0	2	2	0	0	1	0	0	0	0	0	0	0	0	1	0	6	0
15	2	0	4	0	0	0	2	0	0	0	2	0	0	0	0	0	8	0	0	0	0	2	0
16	2	0	0	0	0	0	1	0	2	0	0	0	2	0	0	0	15	0	0	0	0	23	3
17	3	3	2	5	0	2	3	0	3	0	2	0	0	0	1	1	6	0	0	0	0	0	1
18	1	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	6	0	0	0	0	3	0
19	2	5	1	3	0	2	0	0	2	0	2	0	1	0	0	0	11	0	0	0	0	8	0
20	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
21	1	0	4	0	0	1	0	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0	3

Table B.2: Mistakes made in sub-tasks

Appendix C

Consent Forms

C.1 Image rights consent form



Declaração de Consentimento de Direitos de Imagem

No âmbito da realização da tese de mestrado do Mestrado de Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, intitulada **Mobile phone as VR gateway**, realizada pelo estudante Davide António Ferreira Castro, orientada pelo Prof. Rui Rodrigues e sob a co-orientação do Prof. Teresa Matos, eu abaixo assinado declaro que autorizo à filmagem da minha imagem, bem como a difundi-la no contexto de investigação acima mencionado.

A presente autorização é concedida a **título gratuito.**

Porto, __ de ___________ de 20__

____

(Participante ou seu representante)

C.2 Experiment consent form

DECLARAÇÃO DE CONSENTIMENTO

(Baseada na declaração de Helsínquia)

No âmbito da realização da tese de Mestrado do Mestrado Integrado de Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, intitulada **Mobile phone as VR gateway**, realizada pelo estudante Davide António Ferreira Castro, orientada pelo Prof. Rui Rodrigues e sob a co-orientação do Prof. Teresa Matos, eu abaixo assinado declaro que compreendi a explicação que me foi fornecida acerca do estudo no qual irei participar, nomeadamente o carácter voluntário dessa participação, tendo-me sido dada a oportunidade de fazer as perguntas que julguei necessárias.

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

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

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

Porto, __ de ___________ de 20__

____

(Participante ou seu representante)

Appendix D

User Questionnaires

D.1 Pre-experiment questionnaire

Gender

- ☐ Male
- ☐ Female
- ☐ Other
- ☐ Prefer not to say

Age

- ☐ Under 18
- ☐ 18 - 24
- ☐ 25 - 34
- ☐ 35 - 44
- ☐ 45 - 60
- ☐ Above 60

How many times have you tried a VR experience?

- ☐ 0
- ☐ 1 - 2
- ☐ 3 - 10
- ☐ 10+

How confident do you feel about using VR equipment?

☐ 1 (Very unconfident)

☐ 2

☐ 3

☐ 4

☐ 5 (Very confident)

What's your experience with the following VR devices?

	Never used	Used once	Used a few times	Used many times
VR Headset	☐	☐	☐	☐
VR traditional controllers	☐	☐	☐	☐
Hand/finger tracking (through controllers)	☐	☐	☐	☐
Hand/finger tracking (without controllers)	☐	☐	☐	☐

Do you have VR equipment at home?

☐ Yes

☐ No

Do you own a personal smartphone?

☐ Yes

☐ No

How often do you approximately use a smartphone?

☐ Multiple times per hour

☐ Once per hour

☐ A few times per day

☐ Once a day

☐ Once a week

☐ Rarely

How proficient do you consider yourself in handling/operating a smartphone?

☐ 1 (Just the basics)

☐ 2

☐ 3

☐ 4

☐ 5 (Very proficient)

What operating system do you have in your smartphone?

☐ Android

☐ iOS

☐ Not sure

☐ I don’t have a personal smartphone

Which method for navigating the system are you most comfortable with?

☐ Buttons (Recent apps, Home, and Back)

☐ Touch Gestures

☐ No preference

How important do you consider the smartphone to be in your life?

☐ 1 (Insignificant)

☐ 2

☐ 3

☐ 4

☐ 5 (Very important)

D.2 Mid-experiment questionnaire

How confused did you feel using the smartphone in VR?

☐ 1 (Not confused)

☐ 2

☐ 3

☐ 4

☐ 5 (Very confused)

How familiar did the experience of using the smartphone feel?

☐ 1 (Very different)

☐ 2

☐ 3

☐ 4

☐ 5 (Very familiar)

User tasks - Part I

Using the messaging application

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐
The content was well readable	☐	☐	☐	☐	☐
The "copy" feature was easy to use	☐	☐	☐	☐	☐

Using the camera

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐

Using the internet browser

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐
The "paste" feature for the text was easy to use	☐	☐	☐	☐	☐

Taking a screenshot

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the physical buttons felt normal	☐	☐	☐	☐	☐
The devices attached to the phone did not interfere with the task	☐	☐	☐	☐	☐

D.3 Post-experiment questionnaire

How confused did you feel using the smartphone in VR?

☐ 1 (Not confused)

☐ 2

☐ 3

☐ 4

☐ 5 (Very confused)

How familiar did the experience of using the smartphone feel?

☐ 1 (Very different)

☐ 2

☐ 3

☐ 4

☐ 5 (Very familiar)

User tasks - Part II

Using the video application

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐
The content was well readable	☐	☐	☐	☐	☐
The precision of the touch felt the same using the horizontal orientation	☐	☐	☐	☐	☐

Using the shop application

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐
The content was well readable	☐	☐	☐	☐	☐
The keyboard was easy to use	☐	☐	☐	☐	☐

Using the calendar application

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The interaction with the application(s) felt normal	☐	☐	☐	☐	☐
The elements and buttons were easy to select/press	☐	☐	☐	☐	☐
The content was well readable	☐	☐	☐	☐	☐
The keyboard was easy to use	☐	☐	☐	☐	☐

Phone’s screen

Did you have any issues reading information on the screen?

☐ 1 (Never)

☐ 2

☐ 3

☐ 4 (Very often)

Did the touchscreen feel unresponsive/slow?

☐ 1 (Never)

☐ 2

☐ 3

☐ 4 (Very often)

How useful do you think the touch feedback (white dot that appeared on the area touched) was to use the touchscreen?

☐ 1 (Very inconvenient)

☐ 2

☐ 3

☐ 4

☐ 5 (Very useful)

How difficult was it to use the touchscreen, compared to normal out-of-VR use?

☐ 1 (Much easier than usual)

☐ 2

☐ 3

☐ 4

☐ 5 (Much harder than usual)

Hand tracking

Please express your opinion regarding the tracking of the hands

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The position of the hands seemed accurate	☐	☐	☐	☐	☐
The precision of the fingers was satisfatory enough to use the screen	☐	☐	☐	☐	☐
The representation of the hands was useful for using the touchscreen	☐	☐	☐	☐	☐
The presence of the virtual hands damaged the experience	☐	☐	☐	☐	☐
The virtual hands helped to get a stronger connection with the phone and reality	☐	☐	☐	☐	☐
The virtual hands were essential for the use of the phone in VR	☐	☐	☐	☐	☐

Phone's position

Did the position of the virtual phone seem accurate to the real world?

☐ 1 (Totally inaccurate)

☐ 2

☐ 3

☐ 4

☐ 5 (Perfectly accurate)

Did the position of the phone randomly shift in an unnatural way?

☐ 1 (Never)

☐ 2

☐ 3

☐ 4

☐ 5 (Very often)

Physical aspects

How confortable did you feel holding the phone?

	Very uncomfortable	Slightly uncomfortable	Indifferent	Slightly comfortable	Very comfortable
Portrait / Vertical orientation	☐	☐	☐	☐	☐
Landscape / Horizontal orientation	☐	☐	☐	☐	☐

How did the weight of the phone feel?

☐ 1 (Lighter than usual)

☐ 2

☐ 3

☐ 4

☐ 5 (Heavier than usual)

How bothered or restrained did you feel by the device attached on the back of the phone?

- ☐ 1 (Not bothered)
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5 (Very bothered)

Overall experience

Rate your experience with the various components of the experience

	Very bad	Bad	Neither good nor bad	Good	Very good
Position of the phone in the world	☐	☐	☐	☐	☐
Phone's screen visibility	☐	☐	☐	☐	☐
Touchscreen selection	☐	☐	☐	☐	☐
Tracking of hand and fingers position / gestures	☐	☐	☐	☐	☐
Holding the phone	☐	☐	☐	☐	☐

How bothered or restrained did you feel by the device attached on the back of the phone?

- ☐ Yes
- ☐ No
- ☐ Not sure

How bothered or restrained did you feel by the device attached on the back of the phone?

- ☐ 1 (Very unpleasant)
- ☐ 2
- ☐ 3

☐ 4

☐ 5 (Very pleasant)

Final comments

If you have any final comments regarding the tests, please insert them here.

Thank you for participating in this test!