Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática
2016

Luís André Fortuna
Parra Afonso

Interação em Realidade Virtual usando dispositivos móveis

Interaction in Virtual Reality using mobile devices

Universidade de Aveiro
2016

Departamento de
Electrónica, Telecomunicações e Informática

Luís André Fortuna Parra Afonso

Interação em Realidade Virtual usando dispositivos móveis

Interaction in Virtual Reality using mobile devices

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Paulo Dias, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e da Doutora Beatriz Sousa Santos, Professora associada com agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Professor Doutor António Joaquim da Silva Teixeira** Professor Associado da Universidade de Aveiro
vogais / examiners committee	**Professor Doutor Paulo Jorge Carvalho Menezes** Professor Auxiliar da Universidade de Coimbra - Faculdade de Ciências e Tecnologia
	Professor Doutor Paulo Miguel de Jesus Dias Professor Auxiliar da Universidade de Aveiro

agradecimentos

É com muito gosto que aproveito esta oportunidade para agradecer aos Professores Paulo Dias e Beatriz Sousa Santos, orientadores da dissertação, por todo o apoio, disponibilidade e contínuo acompanhamento durante a realização deste trabalho. Ao Sérgio Eliseu pela disponibilização de equipamento e dos modelos 3D que ajudaram ao enriquecimento da dissertação. Agradeço também aos meus amigos pelo companheirismo e a todos os colegas com quem partilhei este percurso académico. Por último, um agradecimento especial aos meus pais, à minha irmã, à minha tia Margarida e restante família pelo apoio incondicional durante todos estes anos.

Palavras Chave Ambientes Virtuais, Interacção, Dispositivos Móveis

Resumo

Tradicionalmente, teclados e comandos 3D como o Wiimote têm sido usados para interagir com Ambientes Virtuais. Apesar de serem simples de usar, dispositivos deste tipo têm algumas limitações: *layout* fixo, necessidade de os utilizadores se lembrarem do mapeamento entre cada botão e função e manipulação de objetos virtuais indireta sendo feita através de botões. Esta dissertação propõe o uso de dispositivos móveis para interação com ambientes virtuais imersivos. Estes dispositivos apresentam algumas vantagens, nomeadamente permitem a existência de interfaces flexíveis e os *touchscreens* juntamente com os sensores internos dos *smartphones* e *tablets* permitem novos estilos de interação, para tarefas como a seleção e manipulação direta de objetos virtuais e navegação. Foi desenvolvida uma aplicação de Museu Virtual para validar o sistema proposto. Com base neste sistema foram feitos dois estudos que abordam duas questões ainda em aberto na área dos Ambientes Virtuais Imersivos: a representação virtual do corpo do utilizador e os métodos de interação a usar para navegação e seleção. Esta dissertação apresenta resultados sobre os efeitos da personificação virtual e mostra que apesar de por vezes não ser tão robusta como o uso de comandos 3D, a interacção usando dispositivos móveis é viável e oferece várias possibilidades de interação interessantes.

Keywords Virtual Environments, Interaction, Mobile Devices

Abstract Traditionally, keyboards and 3D controllers such as the Wiimote have been used to interact with Virtual Environments. Despite being simple to use, these types of input devices have a number of limitations: fixed layout, require the user to remember the mapping between buttons and functions and indirect manipulation of virtual objects, being done using buttons. This dissertation proposes the use of mobile devices to interact with immersive virtual environments. These devices allow to have flexible interfaces and the touchscreen combined with the onboard sensors in smartphones and tablets allow for new interaction styles, in tasks such as direct selection and manipulation of virtual objects and navigation. In order to validate the system a Virtual Museum application was developed. Based on this system two studies addressing two open research questions related to immersive virtual environments: the user virtual embodiment and the interaction methods to use for navigation and selection. This dissertation provides new insights into the effects of virtual embodiment and even if is not always as robust as the use of 3D controllers, the interaction using mobile devices is viable and offers several interesting interaction possibilities.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Virtual Reality (VR) is one of the main technology trends of today. Due to the combination of improved hardware and optimized software and as the latest Head-Mounted Display (HMD) products become relatively affordable to the consumer it is now easier to develop better VR experiences that are much more immersive and realistic. To create an immersive VR environment one of the challenges is how the user will be able to interact with the virtual world. Since there is a lack of standards and constraints 3D interaction can be quite difficult. While designing interaction in Virtual Environments (VE) one must take in consideration three main goals: performance, usability and usefulness. The input technique is of course an important aspect of the interaction. One of the most common options is using input devices such as Wiimotes, Playstation and Xbox gamepads or other similar controllers to perform the universal interaction tasks for VEs: navigation, selection, manipulation ([Bowman et al., 2004]). Although these devices are simple to use, they are also limited in the interaction they allow. Mobile devices like smartphones and tablets by having a touchscreen are not limited to a number of buttons or layout as the mentioned devices which enables a wider ranger of ways to interact with the VE. Another advantage of this kind of input devices is that it is possible to perform a direct interaction and have a highly customizable user interface.

1.2 Objectives

This dissertation intends to determine the extent to which mobile devices (smartphones and tablets) can improve the interaction with immersive virtual environments and whether this type of interaction is more usable and preferred by the users in comparison with using 3D controllers such as the Wiimote. Taking into account this goal one of the aims is to develop applications where the mobile devices are used to interact with virtual environments where we view the virtual world directly on the device screen or on a head mounted display (HMD) as the Oculus Rift. This dissertation also seeks to continue the works by [Souza, 2013] and [Pinto, 2015] in developing a flexible and configurable virtual reality system. The developed application will adapt the imaginary museum concept already used in [Pinto, 2015] that allows to interact with virtual objects in a museum and explore the contents associated with those objects (text, images and videos) as well as adding new virtual objects to the museum exhibit.

1.3 Structure

This dissertation is divided into six main chapters. The first chapter introduces the motivation and objectives behind this work. Chapter two gives an overview of related work and looks at the difficulties and limitations of those works. The third chapter describes the development of a virtual museum mobile application, explaining the main features and interaction methods of the application. Chapter four begins by laying out the design principles of the Immersive Virtual Reality System with tablet-based interaction then describes the architecture of the application explaining the integration and implementation of the major components. The fifth chapter describes the virtual museum demo application based on the developed platform. Chapter six is concerned with user studies and describes the methodology followed by the analysis and discussion of the obtained results in the two usability studies performed, the first concerning the importance of a virtual hand representation and the second a comparison between controller-based and tablet-based interaction in selection and navigation tasks. Chapter seven concludes this dissertation by presenting the main results, the limitations of the developed work and implementation challenges, as well as suggesting possible improvements and future work that may be developed.

Chapter 2

Related Work

This chapter provides an overview of works that use mobile devices to interact in Virtual Reality (VR) systems. A Virtual Reality system may be classified in terms of sense of immersion into two main types: non-immersive and (semi) immersive. In non-immersive system the virtual environment is normally viewed through a standard monitor and as the name suggests the degree of presence is low to none. Semi and fully immersive VR systems aim to immerse the user inside a virtual world giving the feeling that the user "stepped inside the synthetic world" ([Furht, 2008]). The immersion is usually achieved by using multiple projections (CAVE - Computer-Aided Virtual Environment) for semi-immersive VR or by using a Head-Mounted Display (HMD) to have a fully immersive VR system.

2.1 Use of Mobile Devices in Non-Immersive Virtual Reality systems

This section presents works where mobile devices are used to interact with Non-Immersive VR systems and the virtual environment is presented on a standard monitor.
The paper by [Katzakis and Hori, 2010] explores the use of mobile Devices as multi-DOF controllers and starts by indicating that conventional input devices like the mouse and keyboard are not suitable for 3D manipulation tasks. This work also mentions that there are already digital shops where costumers are able to look at products virtually in a 360-degree view. Similarly, in some museums users can interact with virtual content. In such scenarios the use of a mobile device to perform those tasks can be significantly faster and more natural to the user. Since modern mobile devices include embedded sensors such as accelerometers and magnetometers they can work as 3-DOF controllers with a full 360 degrees range in all axis. According to the same authors the use of mobiles devices will allow new interaction techniques. Figure 2.1 illustrates how the touchscreen of mobile devices allow for new interactions like removing a cap of a virtual bottle as we would do with a real one.

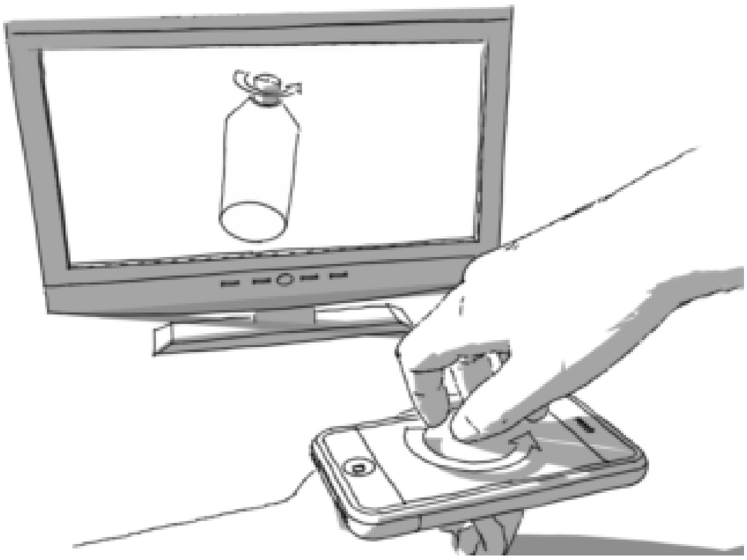

Figure 2.1: Removing the cap of a virtual bottle using a mobile device ([Katzakis and Hori, 2010])

An experimental evaluation compared three different input devices (Mobile Device - HTC Dream, Laser Gaming Mouse and a Touch Panel of a Dell Latitude X2 tablet PC) by asking the participants to rotate one cube to match the rotation of another cube (Figure 2.2). For each input device the subject had to match 10 rotation presets.

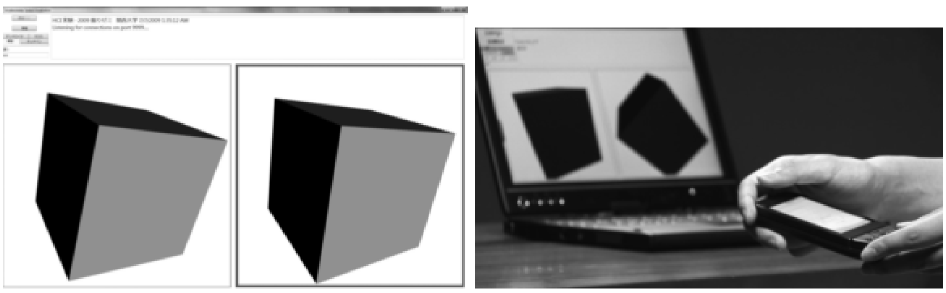

Figure 2.2: Experimental evaluation: rotating an object using a mobile device. On the left: Interface showing the two cubes. The user rotated the cube on the left to match the rotation of the cube on the right. On the right: Experiment setup ([Katzakis and Hori, 2010])

The study by [Katzakis and Hori, 2010] demonstrates that mobile devices can be used effectively as a wireless 3-DOF controller while manipulating virtual 3D objects and that the multi-touchscreens on modern devices can then be used to further expand the range of interactions with the objects. A recent study by [Grandi et al., 2016] also proposes the use of mobile phones for 3D manipulation. Since current smartphones have inertial sensors and a touchscreen which allows to retrieve the device orientation and multi-touch gestures they could be used to perform the three main geometric transformations: translation, rotation and scale. [Grandi et al., 2016] also proposes using collaboration as a way to control multiple degrees of freedom (DOFs). Figure 2.3 shows three users manipulating an object in the virtual environment using Android-based mobile devices. There is a server computer with which the phones communicate via WiFi connection and that is running a Unity 3D application allowing the visualization of the virtual environment.

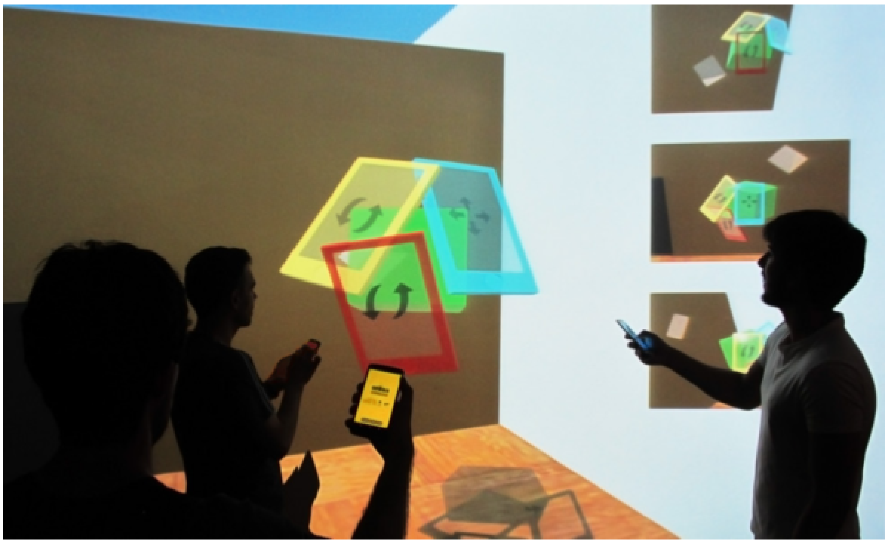

Figure 2.3: Three users simultaneously driving the object through the virtual environment. The colored rectangles indicate the position and orientation of each user in the VE. The three windows at the right-side show the three personal views. ([Grandi et al., 2016])

Figure 2.4 shows all the possible ways of manipulation using the mobile device. Translation is done by touching the device screen and sliding the finger. The object will translate on the plane defined by the phone orientation (Figure 2.4 (a)). For rotation the user has to press and hold the volume down button of the phone and the object will rotate following the orientation of the device (Figure 2.4 (b)). As for the scale it is applied by pinch and spread gestures with two fingers on the device screen (Figure 2.4 (c)). Figure 2.3 shows (on the right) that each user has a personal view of the object. The users can change the camera orientation of this view in a similar way as the manipulation of the object orientation but now by pressing the volume up button (Figure 2.4 (d)).

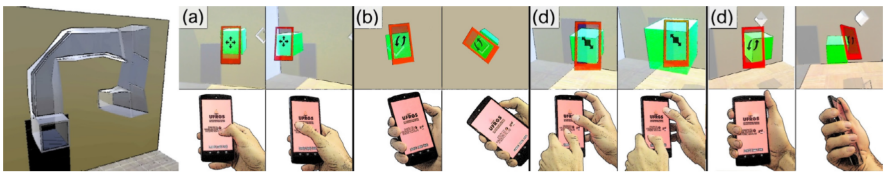

Figure 2.4: Overview of manipulations that a mobile phone cursor can perform over the selected object. ([Grandi et al., 2016])

The solution proposed by [Grandi et al., 2016] simplifies complex 3D tasks by mapping certain touch and orientation gestures performed using a mobile device. This technique can be applied to immersive VR applications that use HMDs, CAVEs or other type of immersive displays. Thus far, a number of studies have tested the efficacy of mobile devices to interact with large-displays ([Bauer et al., 2011], [Graf and Jung, 2012], [Deller and Ebert, 2011]). In this context, an interesting application is the Handymenu ([Lipari and Borst, 2015]), a way to turn a smartphone into a virtual reality controller for menu selection. The touch interface is divided into an area for menu interaction and another one for Virtual Reality object interactions such as selection, manipulation or navigation (Figure 2.5). [Lipari and Borst, 2015] analyzed how the Handymenu compares to the standard ray menus and also what is the best layout to perform menu interactions.

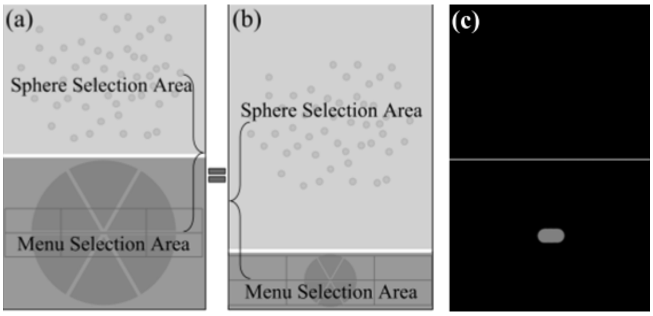

Figure 2.5: Two touch menu size conditions evaluated. A and B: Small and large conditions with relative menu sizes. C: Minimal iPod visuals (menu area center and boundary) ([Lipari and Borst, 2015])

The conducted pilot studies (Figure 2.6) besides comparing the type of object selection (ray and touch) also investigated the types of feedback (touch and visual) as well as the effects of different layouts (pie and grid). Figure 2.5 shows the two touch menu sizes conditions that were evaluated. The participants performed three selections, first a target sphere selection (Sphere Selection Area) and then two menu items (Menu Selection Area). The targets and menus were displayed only on the 3DTV. Selection times, error counts and incorrect touches on the selection area were logged.

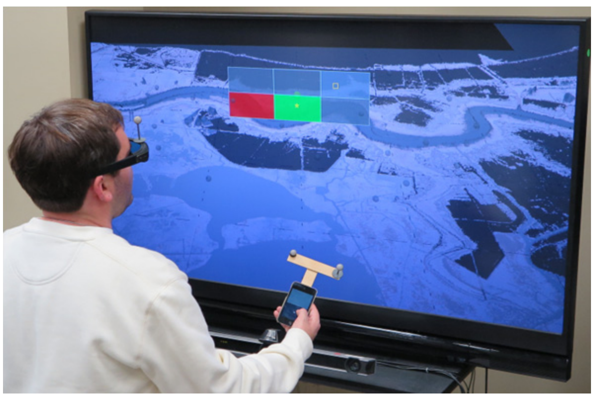

Figure 2.6: Subject using Handymenu in an experiment to compare types object selection and feedback. A 3DTV presented visuals to subjects wearing active stereo glasses and holding an iPod Touch ([Lipari and Borst, 2015])

The touch menu selections were slightly slower than ray-based but the touch selection of menus improved by making the selection area larger.
Besides manipulation and selection tasks, mobile devices have been used for navigation inside a virtual environment ([Bergé et al., 2014], [Benzina et al., 2011]). According to [Bergé et al., 2014] 3D Virtual Environments (3DVE) may be adequate to pass on knowledge in a museum context, however, the traditional used devices such as keyboard and mouse might not be the best interaction devices. In this paper it is suggested that smartphones are a more attractive a stimulating solution for interaction with 3DVE. The authors focus the research on the navigation interaction. Figure 2.7 illustrates the solution designed for the navigation interaction using the smartphone where a) and c) show the actions used to perform translation and rotation respectively.

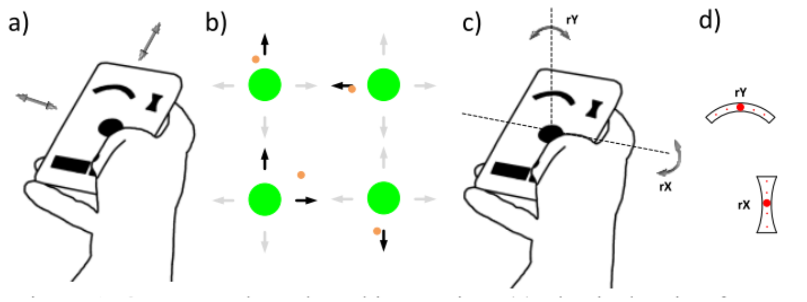

Figure 2.7: The smartphone-based interaction: a) Physical action for translation; b) Feedback of the translation: front, left, front and right, back translation; c) Physical action for rotation; d) Feedback of the rotation. ([Bergé et al., 2014])

An experiment was conducted by these authors to compare the smartphone-based interaction with two common devices: the keyboard-mouse and a 3D mouse (Space-Navigator). The navigation task consisted in moving inside a series of 3D tunnels with a door at the end (Figure 2.8 on the left). In the keyboard-mouse technique (Figure 2.8 a)) the directional arrows were used for translation of the virtual camera and the mouse to change the orientation. In the 3D mouse technique (Figure 2.8 b)) the translation was controlled by applying lateral forces on the device and the orientation was controlled by applying rotational forces. Figure 2.8 c) shows the smartphone technique in use. Since the processing of internal sensors (accelerometer and gyroscope) could be an overload of the smartphone computing capabilities a 6D tracker, Polhemus Patriot Wireless (Figure 2.8 d)) was used and attached to the rear face of the smartphone.

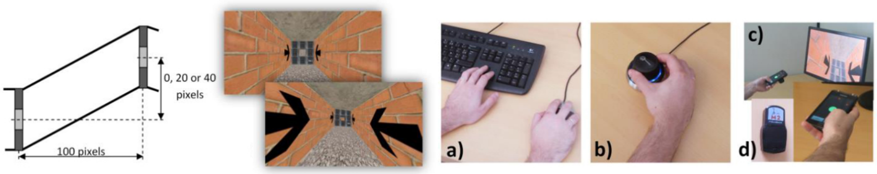

Figure 2.8: Comparing interaction devices. On the left, the 3D environment of the experiment. On the right, the devices used in the experiment: a) keyboard-mouse, b) 3D mouse and c) smartphone with the external 6D tracker, the Polhemus Patriot Wireless ([Bergé et al., 2014])

In terms of results (quantitative and user's preference) the keyboard-mouse technique appeared to be better than both the smartphone and 3D mouse technique. The study has found that after a short learning time the use of smartphones as input device is equivalent in performance to a 3D mouse and with future technical optimization it might become comparable to the use of the keyboard-mouse. Although the smartphone interaction technique still needs improvements, [Bergé et al., 2014] notes that users found this technique more stimulating. Among the works that explore the benefits of using a smartphone as the interaction device in a virtual environment we may cite [Chuah and Lok, 2012] study that consisted in two experiments that used a smartphone as a VR interaction device. The first (Figure 2.9) was a mixed reality game that involves placing an object in a specific location on a game board. The game board consisted on a C# application that showed colored spaces and was displayed by a 22" LCD monitor, placed on a table (Figure 2.9). While performing this task, the user interacted with a virtual human partner (LCD TV mounted to a chair in portrait orientation). The rest of environment consisted of physical game pieces (placed on the nearby shelves) that the user could select and manipulate.

Figure 2.9: Participant using the smartphone to select an object. ([Chuah and Lok, 2012])

When the user selected an object a new user interface would be displayed on the phone screen with an image of the selected object and buttons labeled with the available functions for manipulation (figure 2.10). The smartphone communicated with the game board using VRPN (Virtual Reality Peripheral Network). The components exchanged commands and event messages over a 802.11g network using a TCP/IP socket.

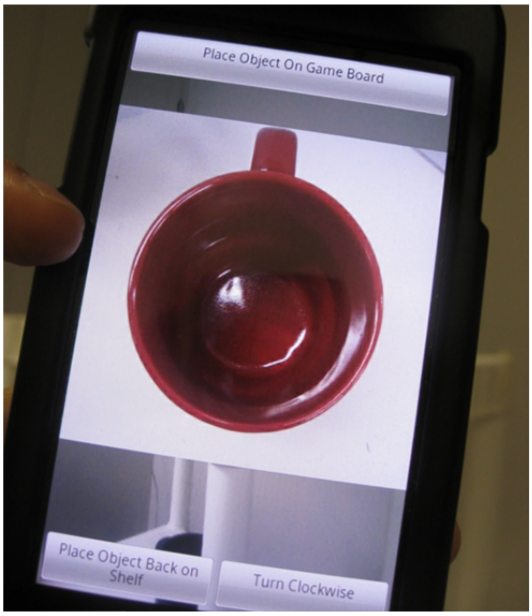

Figure 2.10: Interface for manipulating objects. ([Chuah and Lok, 2012])

The second experiment by [Chuah and Lok, 2012] was an adaptation of an existing application to use the smartphone for interaction purposes. The Virtual Reality Eye Exam is used by medical students to observe symptoms and diagnose cranial nerve damage and symptoms are checked with a series of tests with the patient. This application teaches the users which tests to perform and then how to interpret the obtained results.

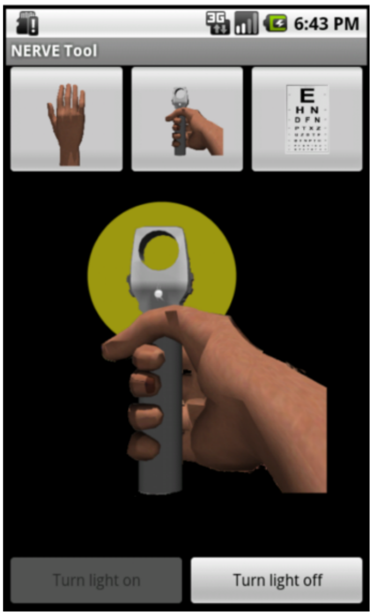

Figure 2.11: Smartphone interface showing the available tools to be used in the virtual eye exam. ([Chuah and Lok, 2012])

The tests are performed using a set of tools and the available tools are displayed as buttons on the phone touchscreen. The large image in the center represents the current selected tool (Figure 2.11). The user is then able to manipulate directly the current tool and perform the virtual eye exam to the virtual human that is displayed on the television (Figure 2.12).

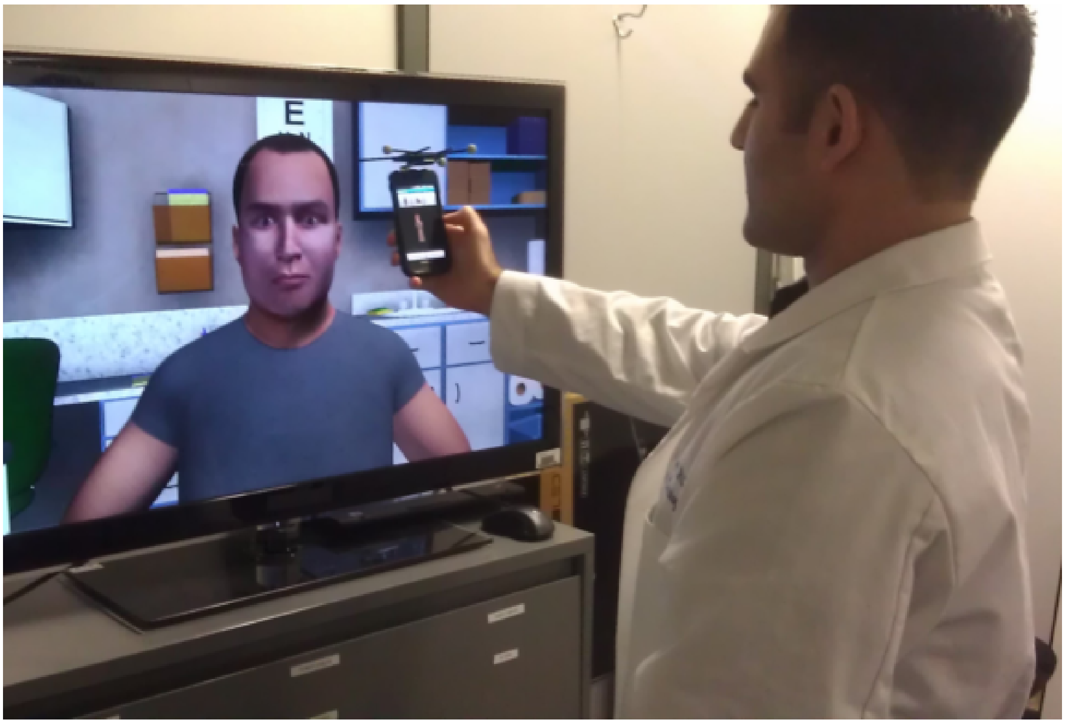

Figure 2.12: Interaction with a virtual human. ([Chuah and Lok, 2012])

Both [Chuah and Lok, 2012] studies suggest that smartphones can improve a range of Virtual Reality and Mixed Reality applications reducing the user need for extra training and button memorization.

2.2 Use of Mobile Devices in Immersive Virtual Reality systems

Turning now to immersive virtual reality systems this section will present works where mobile devices are used as interaction devices in semi (CAVE) and fully immersive VR (using HMD). [Medeiros et al., 2013] propose the use of a tablet-based tool to perform all the main 3D interaction tasks in immersive virtual environments and apply this concept to engineering applications making use of the possibility of mobile devices to display additional information and allow to aggregate all the major universal interaction tasks as navigation, selection and manipulation.

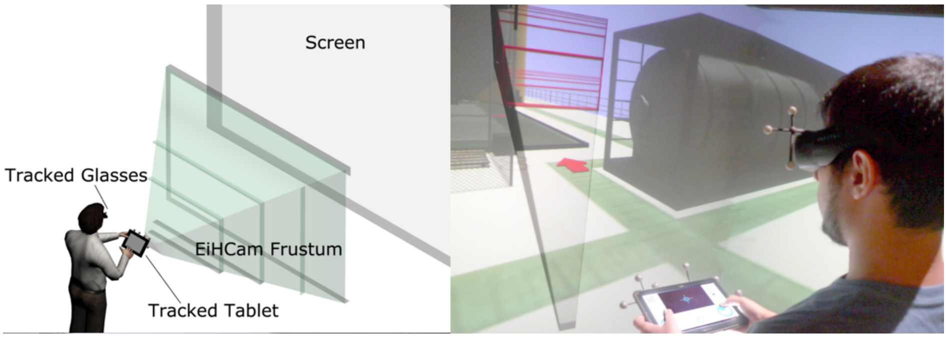

Figure 2.13: Tablet-based tool for 3D virtual environment. On the left: System main components. On the right: User testing the application. ([Medeiros et al., 2013])

In the application proposal the user can navigate using an in-screen joystick (Figure 2.14). The selection technique adapted the use an Eyeball-in-Hand (EiH) metaphor where an extra virtual camera (EiHCam) provides a visualization from the viewpoint of the tablet that the user is holding (Figure 2.14). The tablet is tracked by optical 6 DoF trackers like the ARTracker and BraTracker. As for the manipulation interaction the tablet's touch screen is used to perform selection and manipulation through scale and rotation gestures.
[Medeiros et al., 2013] used a training simulator for oil platforms to validate the developed solution. The participants were asked to perform a set of tasks that use all the proposed features in the developed tool. Figure 2.14 shows the user interface displayed on the tablet and the respective functions of each control.

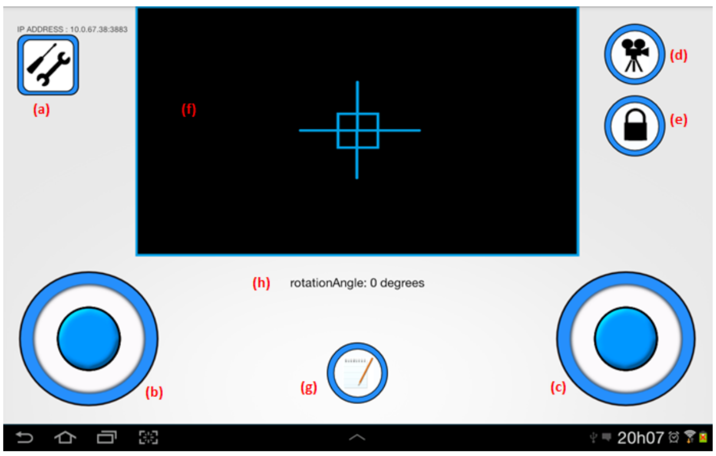

Figure 2.14: Mobile application interface in [Medeiros et al., 2013]. a) configure the IP address; b) and c) navigation controls; d) switch the selection mode; e) lock the EiHCam image or the ray to start the selection/manipulation mode; f) selection (single touch) and object manipulation (double touch) on selected objects; g) add annotation to a selected object

---

Using mobile devices is suggested to be a viable option to interact with virtual engineering content and authors emphasize that this kind of devices are a complete tool for use in virtual environments.

Although smartphones and tablets allow to display a flexible user interface, since the touch-screen is feature-less users cannot easily locate the different virtual widgets without looking at the device screen.[Krum et al., 2014] propose the use of 3D printed panels overlaid on multi-touch mobile devices to add passive tactile feedback. A Unity 3D application runs on the tablet and the interaction mappings can be customizable through software.

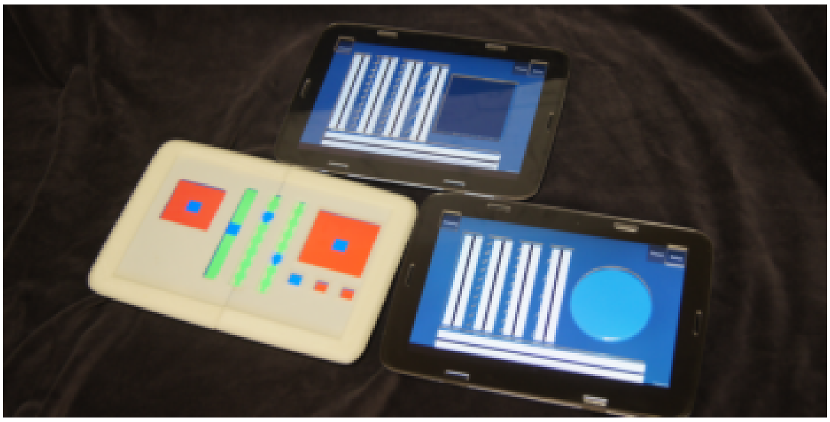

Figure 2.15: Examples of Some Basic Widget Types in Laser Cut and 3D Printed Interaction Panels. ([Krum et al., 2014])

The panels were developed with modern rapid prototyping techniques. A few basic widget types like sliders and square holes that serve as toggle buttons were implemented in the prototypes as shown in figure 2.15. The use of the 3D panels provide tactile and haptic feedback which allows the users to easily locate the controls without the need to look to the mobile device. To aid this type of interaction the heads-up display in front the user shows the current interaction state information of each widget (Figure 2.16).

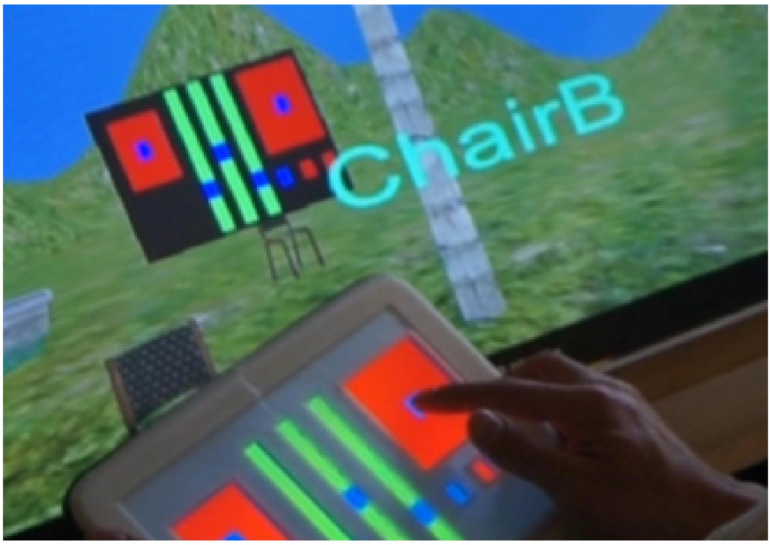

Figure 2.16: Heads-up Display Showing Interaction Panel State. ([Krum et al., 2014])

This system implementation helps the user to interact with a virtual environment using a tablet or smartphone without the need to look at the mobile device. The interaction panel concept can be further applied to a virtual reality context. [Wang and Lindeman, 2015] work presents a new Hybrid Virtual Environment (HVE) metaphor called Object Impersonation. This approach besides allowing to manipulate a virtual object also allows the user to become the object itself and controlling it from inside. In the object impersonation metaphor the user are still able to perform the same 3D User Interfaces (3DUIs) task as in the avatar-base approach and can be effective in several scenarios such as: Remote space inspection (way-finding), Avatar transportation (travel), Occlusion-free object selection (selection), Multi-perspective object manipulation (manipulation) and more. Tablet devices have been successfully used in HVE they since they offer natural bimanual interaction and haptic feeback. These devices can also be combined with spatial tracking sensors to be used as manipulation tools.

Figure 2.17: On the left: The hardware setup of the HVE system; On the top right: user's virtual avatar view on the HMD; On the bottom right: the objects view on the tablet. ([Wang and Lindeman, 2015])

Two object impersonation modes were implemented for object-target alignment tasks: VIEW and DRIVE. In VIEW, the HMD displays the avatar-based model and the tablet screen displays the view of the impersonated object. In DRIVE, the tablet screen displays a third-person view of the virtual object from behind and the HMD displays a view inside the object itself. This DRIVE mode experience is similar to driving a spacecraft from the inside. The new interaction technique proposed by [Wang and Lindeman, 2015] allows to perform 3D interaction from different perspectives and can benefit various 3D UI task scenarios in VR.

The conducted study compared two different object impersonation implementation within a tablet-and-HMD-based HVE system and the results showed that users had a better experience and task performance when using object impersonation combined with traditional 3DUIs. [Saar, 2014] thesis explored new ways to design immersive interaction and navigation in Virtual Reality through a touch device like a Tablet and a Head Mounted Display. Following the investigation of previous interaction techniques [Saar, 2014] tested the effectiveness of a tablet device as the main input device in a Virtual Reality system.

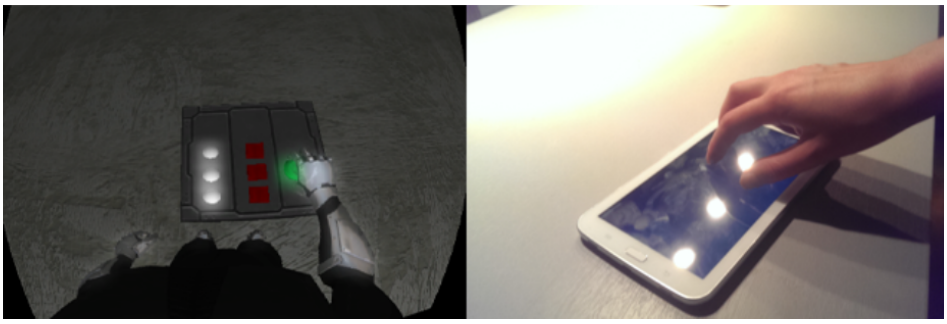

Figure 2.18: Position of the fingers placed in the smartphone replicated to the virtual hand. ([Saar, 2014])

The research main focus was on the navigation interaction and how a touch device like a tablet would increase the user immersion. In the developed prototypes the user was able to navigate by controlling a marker in the virtual world that was itself controlled by moving the finger on the tablet touchscreen. The study suggests that a tablet is a viable interaction device and suited for navigation tasks.

Figure 2.19: User testing the system at home. ([Saar, 2014])

The study also demonstrated that touch devices could work as the main input device in virtual reality and that onboard sensors like the accelerometer allow for the development of natural interactions, however, these devices should be combined with other technologies to allow a wider ranger of new interactions. [Steed and Julier, 2013] work focus on interaction with Virtual Environments using mobile devices' touchscreen as an unseen touch panel in a Virtual Reality system. The main goal of this project was to build a mobile virtual reality system. To achieve this it used an Apple iPhone 4S is combined with a Sony Glasstron LDI-D100BE HMD (Figure 2.20). The iPhone is used to run the system.

Figure 2.20: Fully mobile Virtual Reality system. ([Steed and Julier, 2013])

There is only a small range of available actions (cut, paste, delete). The UI was rendered relatively large on the screen to be legible and add new features in the future (Figure 2.21). The system has limitations regarding the tracking data fusion between the iPhone and the HMD sensors (accelerometer and gyroscope).

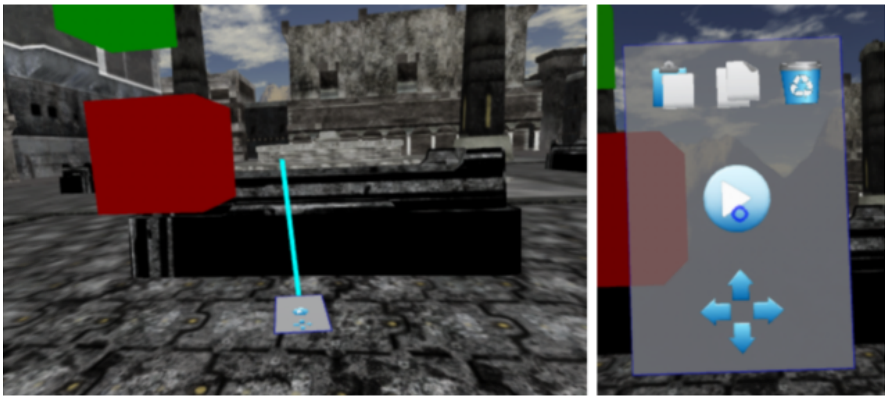

Figure 2.21: System screenshot and virtual panel showing controls for interacting with objects. ([Steed and Julier, 2013])

Despite the limitations this concept presented great potential. The conducted pilot trial showed that this interaction technique was simple to learn and that users had no problem navigating around the environment.

2.3 Conclusion

Overall, these studies provide evidence that mobile devices offer great advantages as interaction devices with virtual environments. When compared to a traditional controller-based interaction there was indication of a reduction of the level of training and memorization required by the user as for example [Chuah and Lok, 2012] concludes the study. [Saar, 2014] besides the new interaction techniques provided by the touchscreen suggests the onboard sensors in the tablets and smartphones like the accelerometer are a great strength of these kinds of devices. Smartphones and tablets can be combined with 3D printed panels to provide tactile feedback ([Krum et al., 2014]) and be used in either non-immersive Virtual Reality (VR), semi-immersive VR ([Medeiros et al., 2013] or fully-immersive VR ([Saar, 2014]).

Chapter 3

Virtual Museum Mobile Application

As described in section 2, it is clear that mobile devices have distinct features allowing different kind of interactions with virtual reality environments compared to a controller-based method. In order to start to explore the use of mobile devices to perform 3D interactions, evaluate current technologies and before the development of a fully immersive virtual reality system a standalone mobile application was developed. This mobile application aims to analyze some of the possible ways to interact with a virtual environment using a smartphone and also to delve into augmented reality as well as into virtual reality with the use of the Google Cardboard.

3.1 Virtual Museum Concept

The Virtual Museum application is based on [Eliseu et al., 2014] Imaginary Museum concept and the developed Imaginary Museum application by [Pinto, 2015]. The Imaginary Museum is a virtual room that replicates a real room from the Museum of the City of Aveiro (Figure 3.1). The virtual room is empty and the main concept of this Imaginary Museum is to allow the users to create their own virtual exhibition by being able to freely place 3D models in the room. There are also several portraits of Aveiro historical figures with which the visitor can interact.

Although the concept of the Imaginary Museum remains the same, the virtual museum mobile application differs from the work by [Pinto, 2015] in a number of respects. Instead of using an Oculus Rift to view the virtual world in this mobile application the virtual environment is viewed directly on the mobile device screen. As for the interactions in [Pinto, 2015] application were mainly gesture-based with those gestures being detected by multiple Kinect sensors in the room (Figure 3.1). In this mobile application the interactions with the environment are now touch-based in combination with the device onboard sensors such as accelerometer and gyroscope. There are two different view modes available. The Regular Mode where only the mobile device is used and the Cardboard mode where the user views the virtual environments through the Google Cardboard, a VR head mount for a smartphones.

Figure 3.1: Room in Museum of the City of Aveiro ([Pinto, 2015])

3.2 Mobile Application

This chapter begins by explaining the graphical engine choice. Then, we describe the two application modes (Regular and Cardboard) and the available interactions followed by the virtual museum features.

3.2.1 Graphical Engine

The first option considered for the graphical engine was the OpenSceneGraph (OSG). This toolkit has support for Android however it is not widely used for development of mobile applications. The installation process caused some difficulties and the available features in OSG Android were not comparable to the ones of the fully featured and popular game engines like Unity 5 or Unreal Engine 4. Both have an active community and an extensive documentation including Android development specific topics. However, Unity offers a better support for third party software. Vuforia SDK was at the time the best Augmented Reality SDK available allowing users to easily create simple to more complex AR applications ([Kim et al., 2014]). It provided a free license with access to the full platform but only offered support to Android (Android Studio), iOS (XCode) and Unity 5 and because of this the choice for the graphical engine automatically fell on Unity 5 rather than Unreal Engine 4.

3.2.2 Virtual Museum Environment

Using 3D models provided by Sérgio Eliseu a scene was set up in Unity similar to the real room of the Museum of City of Aveiro. The main elements here are the portraits of famous Aveiro personalities on the wall and a TV monitor that will be the primary source of interaction with the user. There are two different modes to view and explore the virtual museum environment: Regular mode and Cardboard mode (Figure 3.2).

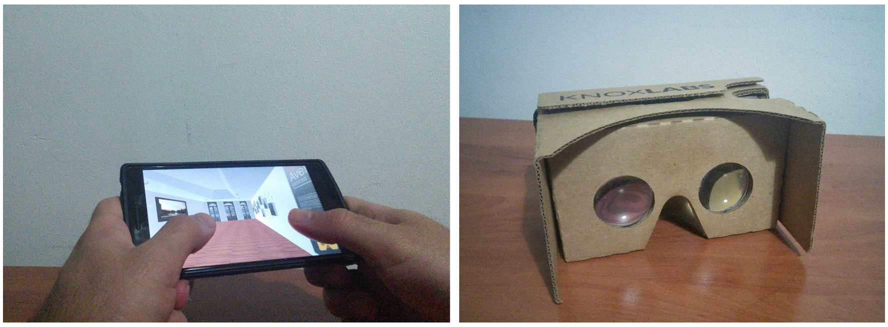

Figure 3.2: Virtual Museum Mobile Application modes. On the left: using mobile application in Regular Mode; On the right: smartphone mounted in the Google Cardboard for usage in Cardboard mode

In **Regular mode** only the Android mobile device is used. Holding the phone the user sees the virtual museum in a first person view (Figure 3.3) through the device screen like any other regular mobile application.

Figure 3.3: Virtual Museum Mobile Application - Regular Mode

The user can look around by rotating the mobile device. The device rotation data is obtained through the gyroscope sensor that are then applied through Unity to the main camera of the application.

In **Cardboard mode** a mobile device is used in combination with the Google Cardboard. The Google Cardboard is a Virtual Reality (VR) head mount for smartphones. It aims to allow users to experience VR in an affordable way. Google provides a Cardboard SDK for Unity [1] with a prefab (a preconfigured Unity GameObject) with essential features like head tracking, stereo rendering already implemented. As in the Regular Mode the user will see the virtual environment in a first person view (Figure 3.4) and can look around by rotating the device.

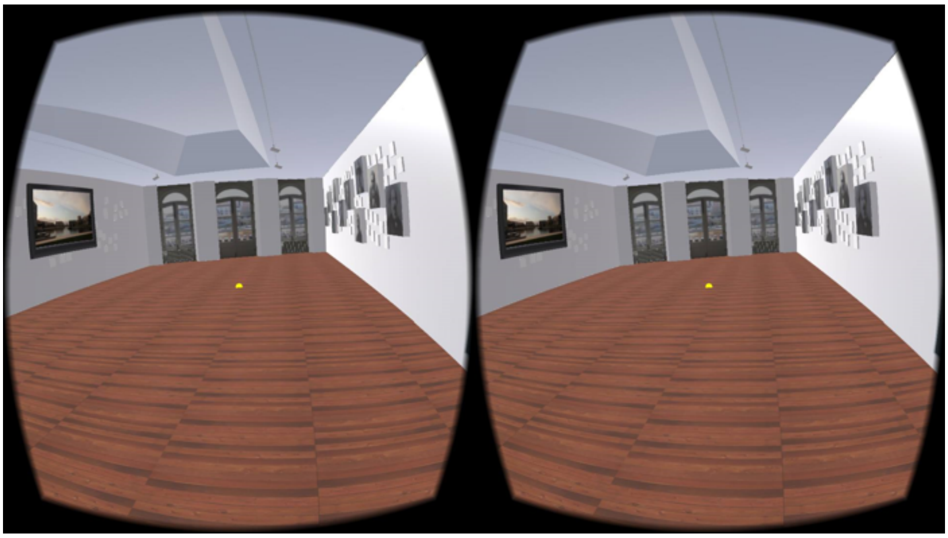

Figure 3.4: Virtual Museum Mobile Application - Cardboard Mode

---

While in Regular Mode the user can change to Cardboard Mode by pressing the cardboard icon at bottom right corner (Figure 3.3). To switch back to Regular Mode the user needs to press the device back button.

3.3 Interactions

This section describes the interaction methods present in the mobile application (navigation, selection and manipulation) considering the two different modes (Regular and Cardboard).

3.3.1 Navigation

For navigation the onboard android device sensors are used. Unity allows to easily access most of the sensors available on the mobile device. The other option was to rely on external tracking devices. The mechanism of the navigation is different in the two available modes (Regular and Cardboard).
In **Regular Mode**, the accelerometer sensor was used to move. While holding the device horizontally by leaning it forwards within a particular threshold the user will move forward in the faced direction (orientation of the camera) and the same to move backwards only now by leaning the device backwards (Figure 3.2).

[1] https://developers.google.com/vr/unity/

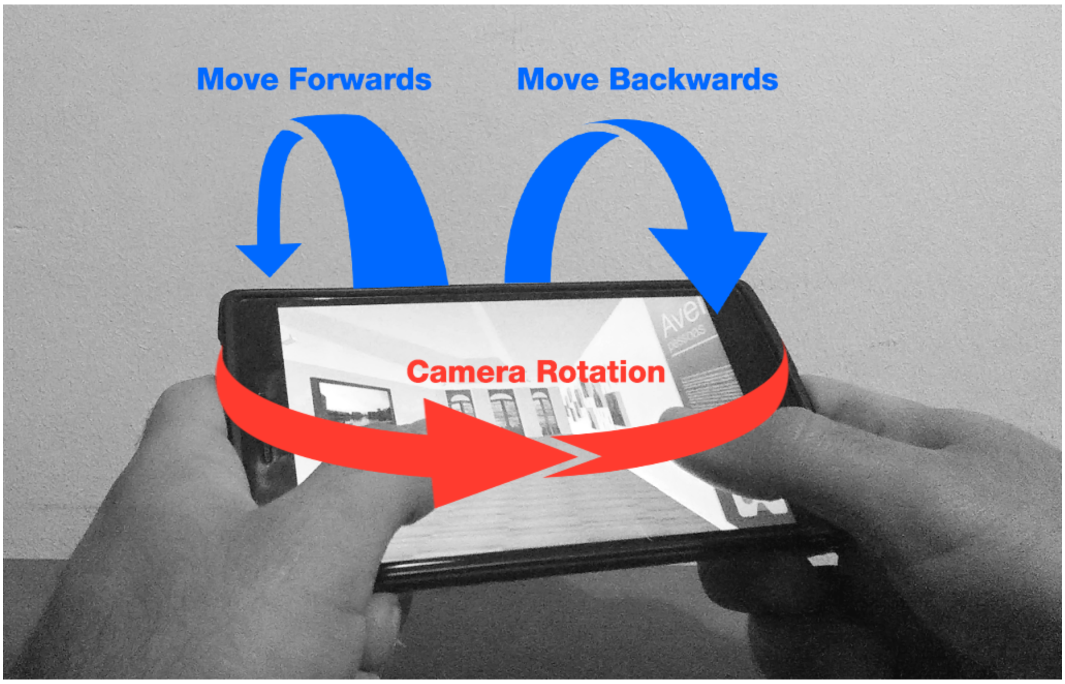

Figure 3.5: Virtual Museum Mobile Application - Navigation method in Regular Mode

In **Cardboard Mode**, the navigation technique developed for the Regular Mode would not really work well. After some investigation we found an implemented solution by Pepwuper Studio[2] for a way to navigate using Cardboard. In this case the user points the gaze cursor of the cardboard to the floor to the desired position to move to (Figure 3.4). Then the user has to press the Google Cardboard button to start the movement.
The implementation by Pepwuper Studio works in combination with the Google Cardboard SDK and the Unity NavMesh class [3].

3.3.2 Selection and Menus

The user can interact with the portraits on the wall in the Museum Room. In **Regular Mode**, performing touch on the position of the desired portrait selects the corresponding personality. In **Cardboard Mode**, the user looks (gaze) at the portrait and the portrait will expand which gives the user additional feedback on the current selection (Figure 3.6). To actually perform the selection the Google Cardboard button needs to be pressed (which is the same as a touch action).

[2]https://github.com/pepwuper/Google-Cardboard-VR-Navigation
[3]http://docs.unity3d.com/ScriptReference/NavMesh.html

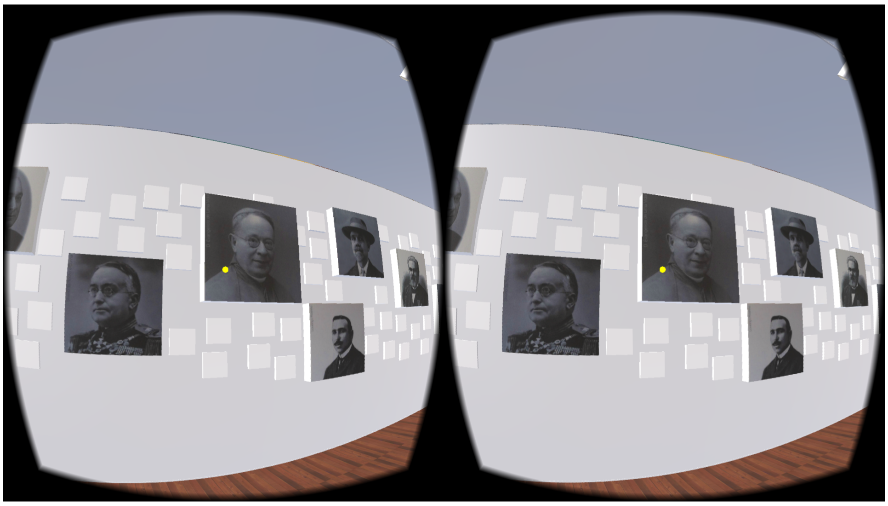

Figure 3.6: Virtual Museum Mobile Application - Gaze Input Selection- Cardboard Mode

After the selection a menu panel will appear with the portrait and name of the selected personality and below a menu with a set of options (Figure 3.7).

Figure 3.7: Virtual Museum Mobile Application - Menu Panel displayed after a portrait selection

To select an option in either mode the process is exactly the same as described in the selection of a portrait. The options in the menu may vary among portraits. The possible options are:

- Biography (Historial). By selecting this option a panel will appear in the center of the screen with a small biography about the personality. This option is available for all portraits.

- Augmented Reality (Realidade Aumentada). By selecting this option the user will go to the Augmented Reality Viewer. This option is only available if there is a 3D model associated to this personality.
- 3D Model (Modelo 3D). By selecting this option the user will go to the 3D Model Viewer. This option is only available if there is a 3D model of this personality.
- Location (Localização). By selecting this option the user will go to the Location Viewer. This option is only available if there is a skybox of the personality monument location associated.
- Quit (Sair). By selecting this option the user will exit the menu and return to the Museum Room. This option is always present.

In addition to being able to interact with the portraits the user can also have the same interaction with a 3D model in the Augmented Reality Viewer or with a 3D model added to the Museum Room.

3.3.3 Manipulation

The manipulation of a 3D model is available in the following scenarios:

- 3D Model Viewer: The camera in both modes is fixed on the 3D Model which can be rotated in any direction by moving the device in Regular Mode or the Google Cardboard in Cardboard Mode. While in Regular Mode the user can also zoom in/out by performing a spread/pinch touch with two fingers and view the statue in more detail.

  In Regular Mode

- Museum Room when adding a new 3D Model: When adding a new statue to the museum the user position and rotation are fixed and leaning the device forwards, back, left or right will instead translate the 3D model.
- Augmented Reality Viewer. The 3D model can be manipulated by moving the printed augmented reality marker.

3.4 Virtual Museum Features

Just like in a real museum the user can freely move around the virtual museum room. As mentioned earlier, the main elements in the room are the TV Monitor and the portraits. The TV Monitor plays videos related to the city of Aveiro. When developing a (PC, Mac or Linux) standalone application we can use a Movie Texture to play a video file. However, Unity 5 does not offer support for Movie Texture in Android. So to be able to play the video in Android we used a solution implemented in the Oculus Rift Utilities SDK for Unity. This solution encapsulates Android Java methods in C# scripts allowing us to call the native media player class of Android. The video playing also caused some performance issues while navigating in the museum with the application being slower.
As for the portraits each one of them has a set of available interactions accessible through a menu and the menu is displayed when a portrait is selected (Section 3.3.2).

Biography (Historial): The biography option is available for all portraits. This option shows a panel with a short biography of the selected Aveiro personality (Figure 3.8).

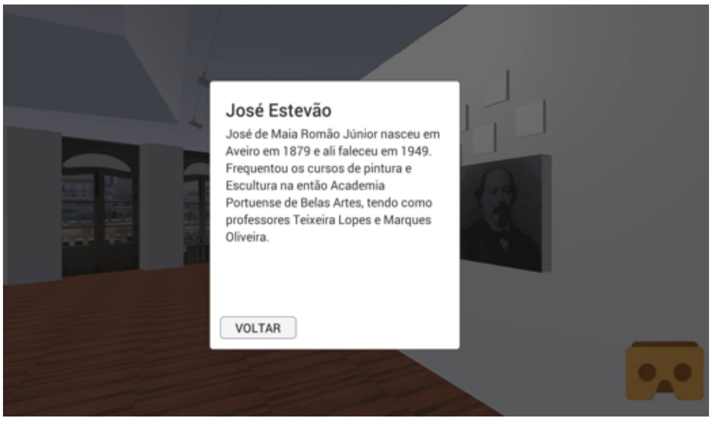

Figure 3.8: Virtual Museum Mobile Application - Biography panel

---

3D Model Viewer: This option renders a 3D model of a real Aveiro Monument of the selected portrait personality. The camera mode used is the same as when the user was in the Museum Room scene. The name of the personality appears at the top center of the screen (Figure 3.9). From here the user also has access to the option to add ("Adicionar") the current 3D statue model to the virtual museum room. If that option is selected the user will return to the museum and will be able to place the statue on the desired position and orientation as described previously in section 3.3.3.

Figure 3.9: Virtual Museum Mobile Application - 3D Model Viewer

---

Augmented Reality Viewer: This is only available while using the Regular Mode. This option renders a 3D model of a real Aveiro Monument of the selected portrait personality given a specific marker (Figure 3.10). Here we used the AR Camera prefab provided by Vuforia SDK [4] for Unity. The natural marker for each personality is the portrait photo. After importing

[4]https://developer.vuforia.com/downloads/sdk

the markers generated at Vuforia Developer Portal into Unity each ImageTarget (a Vuforia prefab) is associated with the respective image/marker and 3D statue model.

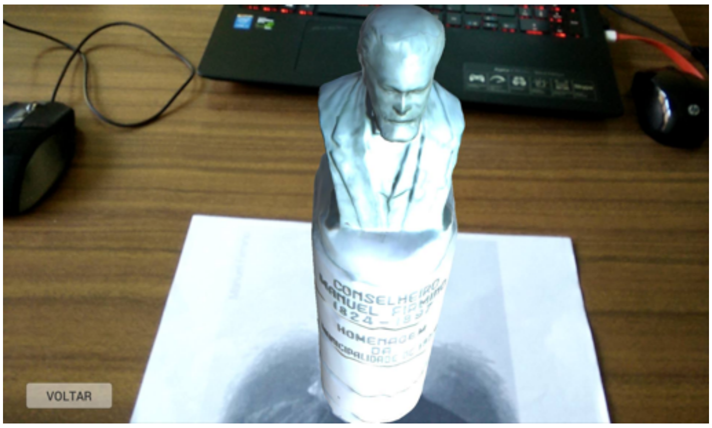

Figure 3.10: Virtual Museum Mobile Application - Augmented Reality Viewer

Location Viewer: Using a free asset called Street View To SkyBox [5] from the Unity Asset Store we generated a 6 sided skybox material with data from Google Street View given the location coordinates. This is simply a scene with a skybox of the real world location of a specific Aveiro personality monument (Figure 3.11). Similarly to when the user is in the virtual museum room it is possible to look around by rotating the smartphone.

Figure 3.11: Virtual Museum Mobile Application - Location Viewer

Augmented Reality inside the virtual environment

In a scenario where the user would visit the real Aveiro Museum room (Figure 3.1) and had a smartphone with the Virtual Museum mobile application it would be possible to either visit both the real and the virtual museum room. With Augmented Reality it would allow to connected these two (real and virtual) visits. While navigating in the virtual museum if the user would point the smartphone camera to one of the portraits on the museum wall a menu

[5] https://www.assetstore.unity3d.com/en/content/47066

would be displayed asking if the user wants to proceed to the Augmented Reality Viewer (Figure 3.12). The AR marker detection is done using the AR camera prefab included in the Vuforia SDK for Unity.

Figure 3.12: Virtual Museum Mobile Application - Augmented Reality Marker detected - Regular Mode

---

3.5 Conclusion

The development of this application extended our knowledge on possible interaction styles when using a mobile device to interact in VR systems. Besides the development of a touch-based interaction the use of the devices onboard sensors like the accelerometer and gyroscope to perform some task was also investigated. All the major interaction methods, navigation, selection and manipulation were explored in a non-immersive context with the Regular Mode and a more immersive virtual reality scenario with the Cardboard Mode. This application serve as a base for the development the immersive VR system with tablet-based interaction presented in the next chapter.

Chapter 4

Immersive Virtual Reality System with Tablet-Based Interaction

This chapter starts with an overview of the developed system describing the design principles and architectural decisions then proceeds to explain the major aspects of the implementation process.

4.1 System Design

Before proceeding to explain the details of the system implementation, it is important to determine exactly what is the system purpose, the design principles in which is built on and explain the main architectural decisions made.

4.1.1 Design Principles

This system is an immersive virtual reality system that can be easily configured and where a tablet device is used to perform interactions with the virtual content. It is based on the following two principles: **configurable virtual environment** and **tablet-based interaction**. In recent years, previous works have begun to develop systems for setting up configurable virtual environments ([Souza, 2013], [Pinto, 2015]). In pSIVE [Souza, 2013] main motivation was to decrease the level of expertise required to construct a virtual immersive environment. The system allowed to customize the models in the virtual scene as well as the associated content like text information, PDFs or videos without the need of programming skills. The framework developed by [Pinto, 2015] added new features to pSIVE ([Souza, 2013]) and a new interaction method through gesture input. Support to the latest generation of Head-mounted displays (HMD) was also updated. Together these master theses provided important insights into the customizable virtual reality systems. This project keeps as one of his objectives to develop a system for easily setting up virtual environments.
As discussed earlier in chapter 2, there are already some authors who showed the effectiveness of smartphones and tablets as an unseen interaction device in a virtual reality system. The use of game controllers like the Wiimote as input devices as well as gestures-based input have been tested in [Souza, 2013], [Pinto, 2015] frameworks. Although the results from the conducted experiments in both works were mostly positive those input methods do require some training from inexperienced users. While the controller-based input only provides indirect interaction,

the gesture-based input may be more similar to the way we would manipulate objects in the real world but since we are not actually holding the object there is no tactile feedback thus reducing the sense of immersion and realism. The tablet device touchscreen allows a different range of interactions and visualization of the virtual content. The applied concept is similar to the implemented "pen & tablet" metaphor by [Bowman, 1999]. The physical tablet device and the user hands are both tracked and have a graphical representation in the virtual environment. The 2D interface is displayed on the virtual tablet screen in which the user interact by touching. A significant advantage is that the interface is always available therefore always accessible and can be conveniently moved so it does not block the environment. Most of us are now used to interact with touchscreen be in smartphones, tablets or even laptops and so the tablet-based interaction is expected to be natural and easy to learn thus improving the immersion in a virtual environment.

4.1.2 Graphical Engine

The graphical engine used for the development of this project is obviously an import element of the system. The key aspects taken in consideration while choosing the graphical engine were:

- Support to the used hardware and third party software;
- Ease of learning;
- Ease of use;
- Community activity;
- Quality of documentation.

Similarly to what was mentioned in section 3.2.1 (Graphical Engine) regarding OSG Android the 3D Engine OpenSceneGraph (OSG) is currently outdated and is not comparable to a game engine, in terms of features and support for third party software. The game engines Unity 5 and Unreal Engine 4 were the two main contenders. Both engines provide high quality documentation as well as an active and helpful community. However, since Unity had a free version long before Unreal Engine there were at the time more available tutorials and explanations about aspects of the development process not only from official sources, but also from professional and hobbyist developers around the web. For the same reason a lot of third party software already offered direct support to Unity via .unitypackage which is the case of the Vuforia SDK which provides augmented reality features essential to this system. Unity 5 was also already used during the development of the mobile application thoroughly described in chapter 3 so the author already had some experience with this game engine. Unreal Engine 4 has a visual scripting system that can be used in combination with C++ scripting whereas in Unity 5 the scripts are usually programmed in C# with which the developer had a higher level of expertise compared to C++. For these reasons the decision ultimately fall on Unity5.

4.2 System Architecture and Implementation

Having delineated the design principles and the graphical engine that will be used for the development, this section describes the system architecture including how all the involved components were integrated and implemented in the solution.

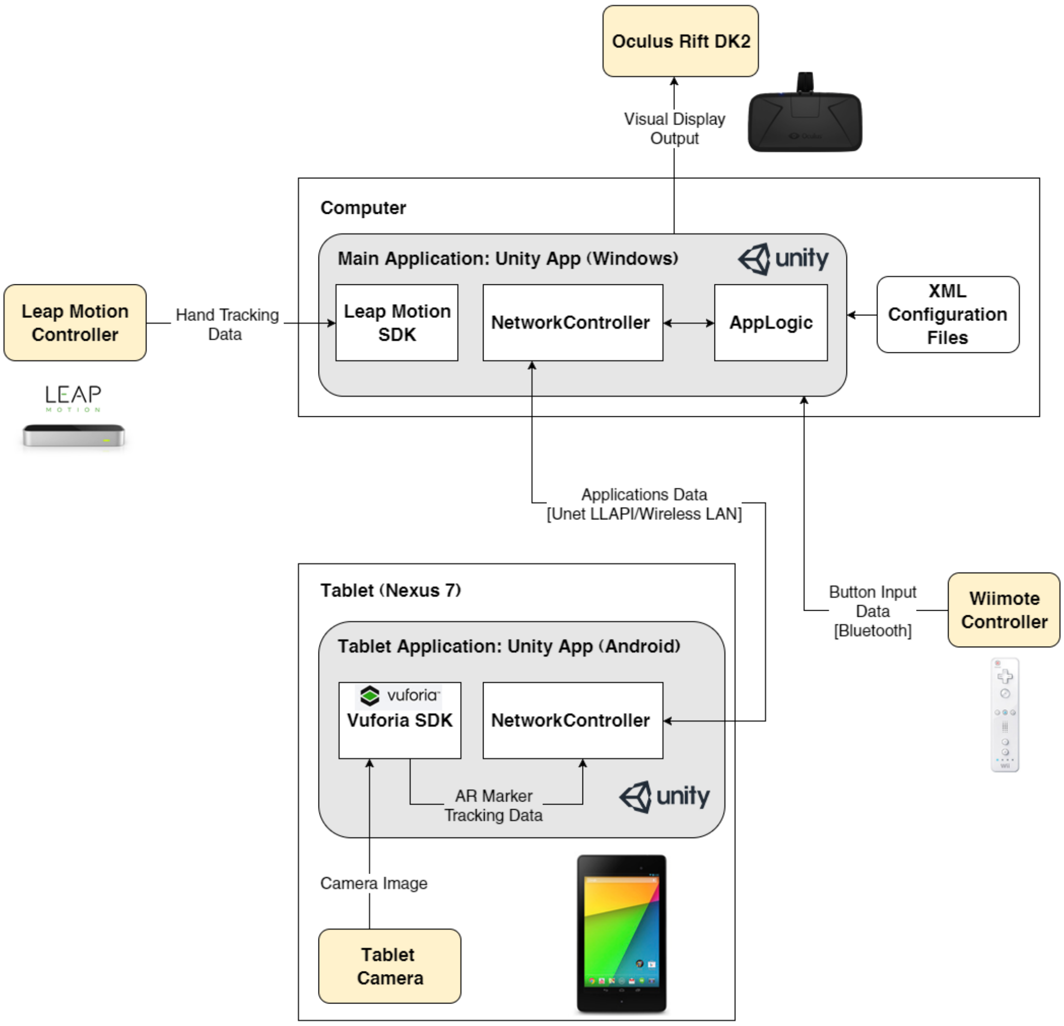

Figure 4.1: System Architecture

The system architecture detailed in figure 4.1 gives an overview to how all the hardware and software components are connected. The tablet and the Wiimote are available interaction devices. They work independently, the user will only use one of the devices to interact at a time with the VR (main) application running on the Computer. Both support the navigation and selection methods. Each of the components presented in the system architecture (Figure 4.1) will be explained in detail as well as the implementation aspects such as the integration of the hardware components and communication between the components.

4.2.1 Main Application

The computer connects all the hardware components and the Unity application (Main application) running on the computer plays a similar role by connecting the software components. The hand tracking data sent by the Leap Motion controller connected to the computer is retrieved in Unity by the **Leap Motion SDK**. The SDK is installed on the Unity project via an Unity extension provided by Leap Motion for developers. The SDK also provides Unity

hand prefabs ready to be added and used in any Unity scene. As for the **visual output**, Unity (from 5.1 version onward) already has built-in support for VR devices including the Oculus Rift DK2. This means that when the virtual reality option is activated in Unity and the VR headset connected to the computer the output will be rendered in stereo and the head-tracking input automatically applied.
The two key C# scripts in the main application are the AppLogic and the NetworkController. The **AppLogic** script controls the main application internal logic. It receives data from the NetworkController and other minor scripts associated with Game Objects in the current scene and acts according to the programmed logic. The role of the **NetworkController** is to exchange messages with the **tablet application** related to both applications information. This task is facilitated by Unity's Unet Transport Layer specifically Low-Level API (LLAPI) which allows for the creation of our own network system. The LLAPI works as a sockets-based networking system. It is possible to send and receive messages and choose QoS (Quality of Service) options according to the system requirements. The setup is exactly the same as any socket networking it is necessary to first initialize the Network Transport Layer, configure the network topology and create host. The communication starts when the connection between the main application and the tablet application is established. A socket address is the IP address plus the port number. The port numbers are already setup up in the script but the IP addresses of each machine (tablet and Computer) can be set up on the start up of each application. The messages sent are no more than simple commands (strings) with data of the tablet application like indicating what button was pressed and AR tracking . The main framework after receiving the commands will execute an action according to what is defined in the AppLogic script. Depending on other interactions within the main application commands could be sent to the tablet application to change the interface shown.
As explained earlier, one design principle on which the platform is built upon is to allow the virtual content to be easily configurable. A simple way to define configuration for the developed system is by using the XML markup language. This format has the particularity of besides being machine-readable is also human-readable. The **XML configuration files** are read by the Unity application at the start-up via a developed script (C#) and the configuration are applied to the respective Game Objects. An example of a XML configuration file is described on section 5.2.
Unity does not provide an out of the box feature to access the **button input data** of the Wiimote but this problem was easily solved with Unity Wii Remote API developed by Flafla2 [1]. This API allows to get all the necessary data from the Wii Remote. The installation process is always the same when using external libraries we simply have to import the provided Unity package. The button input was then associated with specific actions in the virtual environment as described in figure 4.2.

4.2.2 Input and Output Devices

The system allows for several interaction devices. In our work we focused on: the tablet (Nexus 7) and the Wiimote. In this context we want to compare the effectiveness of tablet-based interaction against controller-based interactions a more common way to interact in immersive virtual environments. So an implementation of the platform with Wiimote as interaction device was developed.

[1] https://github.com/Flafla2/Unity-Wiimote

Tablet: When using the tablet to interact with the system the tracking of the device position and orientation is done using augmented reality as will be explained later in detail in section 4.2.4. A typical augmented reality (AR) setup needs two elements, a camera and an AR marker. In this system the **tablet front-camera** is used to detect the AR marker placed in the front of the Oculus Rift (Figure 4.5). The AR marker is a printed out image chosen by the developer. For the image to be detected by the software (Unity/Vuforia SDK) it needs to be previously added to the Vuforia Target Image database where then it is possible to export as a Unity Package and used in Unity.
The **tablet application** running on the Tablet exchanges information with the Main application. With the **Vuforia SDK** we obtain the tracking data that is sent in a message in every Update call (Update is called every frame). Whenever there is a touch on the screen or on a specific UI button this information is also sent to the main application. The **NetworkController** script plays a similar role to what already described **NetworkController** script of the main application (Section 4.2.1).

Wiimote: Traditionally in an Immersive Virtual Environment (IVE) game controllers are used as input devices. They are simple to use and easily available. The buttons of the controllers can be mapped to perform specific interactions with the Virtual Environment (VE). As ([Chuah and Lok, 2012]) states this type of devices generally work well in these scenarios but require the user to remember the functions associated to each button of the input device. This first implementation of the framework using the Wiimote as the interaction device will serve as reference point for later comparison in usability studies with the tablet as an interaction device. Other input device such as the Playstation Move ([Takala et al., 2013], [Takala and Matveinen, 2014]) or the Xbox Controller ([Zielinski et al., 2014]) were also viable options but the Wiimote was chosen since is more commonly used for 3D user interfaces ([Wingrave et al., 2010]) and was available at the time. Figure 4.2 shows the mapping between button and function. Only the D-pad and 'A' button were used to perform interactions.

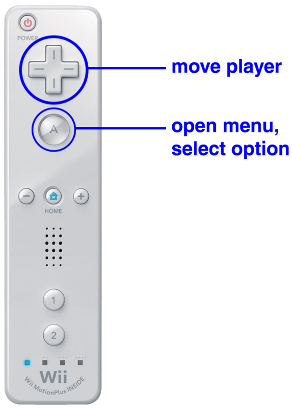

Figure 4.2: Wiimote button - function mapping

The **Oculus Rift** Development Kit 2 is an Head Mounted Display (HMD) used to display and the most popular Virtual Reality headset. The Rift has 6 DoF but our application only makes use of 3-axis rotational tracking. It needs to be connected to a computer with the proper Oculus Runtime drivers installed.
The **Leap Motion controller** is a device connected to the computer via USB that pro-

vides hand and finger tracking. The recent updates of the Leap Motion SDK main focus was to improve performance in Virtual Reality. Although the tracking capabilities keep getting better there were still some problems with the hand tracking specially while holding real world objects like a tablet. The role of the hand tracking in the developed platform is to provide additional visual feedback to the user while using the tablet-based interaction in and immersive virtual environment (IVE). The benefits of virtual embodiment have already been explored by some authors. [Ries et al., 2008] investigated the effects of a self-virtual avatar in estimation of egocentric distances where the results showed that users performed better while having a virtual avatar. In a recent study, [Steed et al., 2016] argue that a self-avatar may reduce the cognitive load in an IVE.
Figure 4.3 shows all the hardware components that are part of the system. The computer running the main application is one of the major components since it connects all other hardware components. The Oculus Rift is connected to the laptop via USB and HDMI for the video output. The Leap Motion is connected also via USB port. As for the interaction devices, the Wiimote controller exchanges the button input data with the computer via Bluetooth. The Android tablet device, running the tablet application, works over a wireless LAN connection. The interaction is either controller-based (Wiimote) or tablet-based (Nexus 7 tablet).

Figure 4.3: Hardware setup. 1) Computer running the main application; 2) Wiimote controller; 3) Tablet (Nexus 7) running the tablet application; 4) Leap Motion Controller connected to the computer; 5) Augmented Reality marker attached to the Oculus Rift DK2, Oculus Rift DK2: virtual reality headset, connected to the computer

4.2.3 Interaction Methods

This subsection describes the available interactions in this platform for the two interaction devices, Tablet and Wiimote.

Navigation

Realism in Virtual Reality is not always better. A well designed low-fidelity locomotion technique can have excellent performance and be as effective as a high-fidelity locomotion technique ([Nabiyouni et al., 2015]). In this system three different low-fidelity navigation techniques were developed two of them using the Tablet and the other with the Wiimote.
Tablet: The taxonomy for travel techniques developed by [Bowman et al., 2004] states that the task of moving control can be divided essentially in two parts: indicating position and indicating orientation. The orientation can be controlled with gaze steering, in other words controlled by where the user is looking at when wearing an HMD. As for the position, there are two different techniques that can be used to move in the virtual environment. In the first method the user can control the movement with a virtual joystick displayed on the tablet touchscreen. The virtual joystick will be active each time a touch action starts. While touching the tablet screen, the user can move the finger into any direction. The speed can also be controlled and it gradually increases according to how far a continuous touch is from the initial touch position. The virtual joystick used was from the Unity asset developed by Cyrill Nadezhdin [2]. The second way to move is via teleport together with the first method with the virtual joystick. In the bottom half of the screen the technique is the same as the previous one, the user can drag the finger around that area and move into the desired direction. In the top half the screen the mini-map representing the virtual environment will be displayed in the virtual tablet where the relative position of the user will be marked in that mini-map. By clicking in the mini-map area is possible to teleport to that location in the virtual environment.
Wiimote: In the controllers-based interaction (Wiimote) the navigation method works using the D-pad buttons to move (Figure 4.2). The direction of the movement to the direction where the user is looking.

Selection and Menus

Tablet: The selection interaction may be divided into three main sub tasks: feedback, indication of object and indication to select ([Bowman et al., 2004]). In our system, pointing the virtual tablet that represents the tracked tablet hold by the user performs the indication of the object. When pointing to an interactable object in the virtual environment there are two kinds of graphical feedback. The object in question will slightly expand and information about the object will be displayed in the virtual tablet screen (Figure 5.2). To execute the selection (indication to select) a virtual button (Select) presented in the 2D interface of the tablet needs to be pressed (Figure 5.2). If the selection is successful the interface will change to a menu with multiple buttons with mapped functions that allow the user to interact with virtual content associated with the selected object (see Figure 5.3). The menus displayed in the 2D interface will be of a linear type which seemed to be the most adequate to the interaction device used. When pressing a menu option, the button will be highlighted (color

[2] https://www.assetstore.unity3d.com/en/content/15233

changed) to provide graphical feedback to the user.
Wiimote: Similarly to the Cardboard Mode in the Virtual Museum on Tablet application the gaze input is used to choose a portrait. To select that portrait instead of pressing the cardboard button the user now needs to press the 'A' button in Wiimote. After the selection, the menu appears floating in front of the user (see left image in Figure 6.7). The gaze input will also be used to choose one of the available options and to select the pressing of the cardboard button will be replaced by the 'A' button in the Wiimote.

4.2.4 Tablet Tracking

The main limitation of ([Steed and Julier, 2013]) tablet system was the tracking. Over time, there is a divergence of values between the head and the hand sensor. One of the major difficulty of using a tablet as an interaction device in our system was also the tracking. What follows is a description of the methods evaluated in our system for effective tracking.
The first approach to accurately compute the position and orientation of the tablet was to use **WinTracker III**, a 6DOF magnetic tracker. Using one of the receivers attached to the back of the tablet (Figure 4.4) and by sending the tracker output (Cartesian coordinates of position and Quaternion orientation) to Unity it was possible to replicate in the virtual world the real world position and orientation of the tablet. This data is not directly sent to Unity. Virtual Reality Peripheral Network (VRPN), a library that implements a network-transparent interface between the program and a tracker, was used. It is a commonly used library and already supports a wide range of hardware devices (including Wintracker III) but also allows to create device classes for trackers that are not currently supported. Once Wintracker III is connected to the PC via USB it is then necessary to start the VRPN server. On the start up VRPN server reads a configuration file with information about what devices will be active, the device name identifier and for each device what sensors will be active. Wintracker has 3 receivers and the Update Rate changes depending on how many receivers are active. In our case only one receiver was used (Figure 4.4).

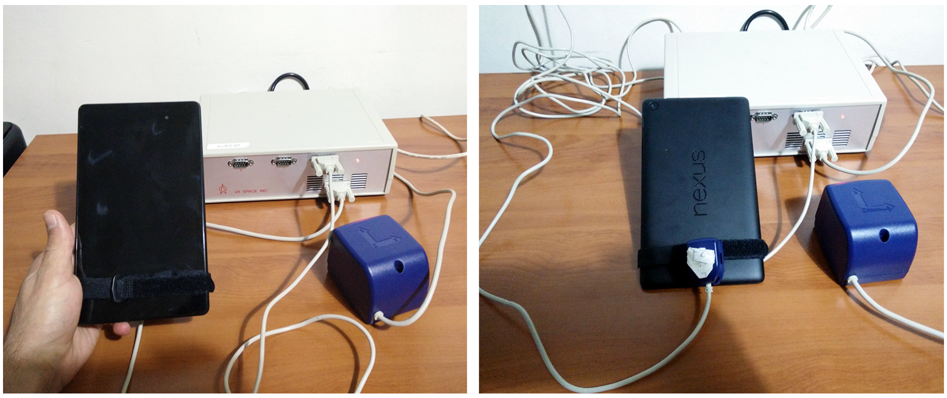

Figure 4.4: WinTracker III - Virtual Reality Tracker attached to the Nexus 7 tablet

---

Unity does not offer direct support to communicate with the VRPN server. A VRPN wrapper for Unity developed by Scott Redig [3] was used for this task. A simple script for

[3]https://github.com/Laremere/unityVRPN

Unity was then developed to apply the position and rotation component of the tracker obtained through the methods provided by the wrapper to the virtual model of the tablet. It is only necessary to set beforehand the VRPN server address (e.g., Wintracker0@localhost) in which the VRPN server is running.
Although WinTracker III is relatively old device it does have a decent update rate (90 outputs/second with one receiver) and latency (7ms) as well as good accuracy (Static Accuracy is 0.06" RMS for position axis and 0.3° RMS for orientation variables. Being a magnetic tracker there was some interference while reading data with the receiver attached to the back of the tablet. By re-positioning the WinTracker receiver in lower end of the tablet did decrease the effect of the interference. The use of a more recent tracker like the Sixense STEM system would possibly offer an even more precise tracking and being wireless would increase the freedom of movement of the user however in our application the expected area of interaction rather small. Using a tracker attached to the tablet does solve the problem of getting the position and orientation components however regardless of the tracker used it is always an additional device apart from the HMD and the tablet which turns the replication of this application more difficult and costly. Another solution is to get this data without resorting to an additional physical device.
Traditionally, a simple Augmented Reality (AR) use-case is to associate to a 3D Model to a marker and when pointing a camera to that specific marker show that 3D Model. This is actually the same AR scenario used in the Virtual Museum on Tablet application described in chapter 3. Using the Vuforia extension for Unity it is easy to get the position and orientation of the tablet camera relatively to the AR marker. With this data it is possible to map the tablet position in the virtual world.

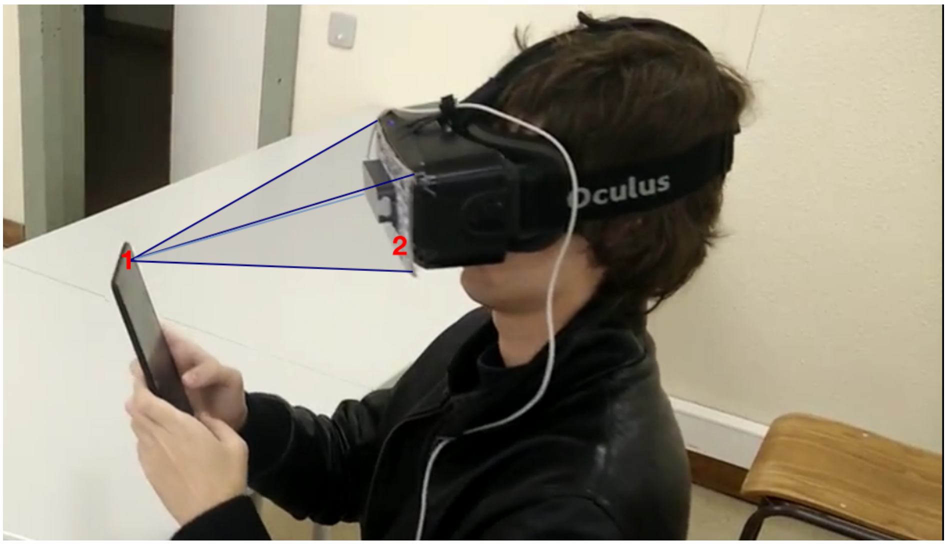

Figure 4.5: Augmented Reality used to track the tablet device. 1) Tablet front-camera that will track the AR marker 2) on the front of the Oculus Rift.

This **AR tracking** is comparable in accuracy to the use of a physical tracker like the WinTracker in the first approach. There is no extra device attached to the tablet we only need its camera pointed to the marker to perform the tracking, however, as mentioned previously the area of interaction is small. The user will be seated while using the application and the

tablet is expected to only be moved around within less than one meter radius around the user. The approach taken was to stick an AR marker on the front of the Oculus Rift DK2 and use the tablet front-camera to perform the detection (Figure 4.5). This ensures that when the user is looking directly at the tablet screen the device camera is also pointed to the marker on the Oculus and we are able to know the position and orientation of the Oculus relatively to the tablet front-camera and consequently we know the position and orientation of the tablet relative to the Oculus. When the user is not looking directly to the tablet the device front-camera is not able to see the AR marker on the Oculus, however, this is not really a problem since if the user is not looking at the tablet it does not matter if the tablet is being tracked. This tracking solution can be robust when following the best practices and recommendation by Vuforia. One of the recommendation is that the image target (AR marker) used should be between 4 and 5 stars rating. This rating can be viewed after the developer uploads the desired image target. The stars rate show how accurate the tracking will be. The image should also have good contrast and be rich in detail and Vuforia SDK tracking also works best in indoor environments with proper lighting conditions. If for some reason there are tracking problems there are two ways to tackle this issue. Use the Extend Tracking capability of Vuforia that uses features of the environment as a way to improve and continue the tracking even when the marker is no longer visible to the camera. A second way is by using the tablet on board sensors like the accelerometer and gyroscope making it possible to estimate the position and orientation values despite some drift over time. The image target/AR marker can be easily customizable when using Vuforia and just needed to be printed out on regular paper which is a great advantage when compared to an external tracking device both in terms of set up and cost making a lot easier for anyone to use this tracking mechanism.
The **Leap Motion sensor** was integrated in the platform with the purpose to track the user hands' movement but we also explored the possibility to use this sensor to track the tablet. The Leap motion uses infrared stereo cameras as tracking sensors (Figure 4.6). In the latest Leap Motion SDK developers have access to the images from these cameras.

Figure 4.6: Image from the Leap Motion infrared stereo cameras.

An idea that arose was to use these cameras to detect AR markers and track the tablet in the same manner described previously. However, the Vuforia SDK for Unity did not permit to define the Leap Motion controller as the detection camera. Besides hand movement the Leap Motion can also detect tools. The classification of tool in the Leap Motion software is given to a pointable object like pens, pencils or a rolled paper (Figure 4.7).

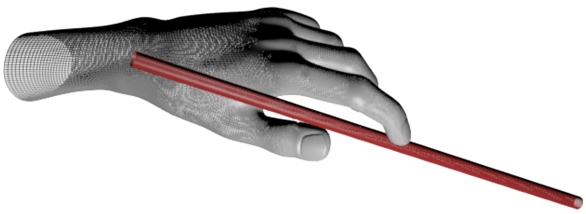

Figure 4.7: Representation of an object detected as a tool by Leap Motion software

Given this functionality another way to track the tablet was tested. By attaching a tool to the tablet and since we know the tool position and orientation it we could also know the location and orientation of the tablet. It is an interesting way of tracking that also would not require additional device specifically for the tablet tracking but from the tests made the tracking was not as effective as the Augmented Reality tracking described before.
During the development of the system a **Google Tango Tablet** was available for testing (Figure 4.8). The development kit released in 2016 is a really powerful device running Android that combines 3D motion tracking with depth sensing. It can also recognize the world around correcting possible errors in motion tracking by recognizing areas seen before. This kind of features in one single device certainly allow to explore new application concepts. Google provided an Unity extension so it was easy to start developing for this device.

Figure 4.8: Google Tango Tablet Development Kit.

In this application scenario the problem to solve was how to know the position and orientation of the tablet device relatively to the user. The Google Tango already knows for itself its position in the real world so what was missing was to know the Tango position relative to the Oculus (user). To do this it would be necessary for each different user to perform some kind of calibration to know the start position of the Game Object that would represent the tablet. This concept was tested, but we encountered some problems while applying it to the main application. The Google Tango lost tracking multiple times due to lighting conditions and some times it was not able to re-calibrate and correct its position being necessary to restart the application.

Chapter 5

Virtual Museum Application

In order to validate the developed system, the virtual museum mobile application (Chapter 3) was adapted to a fully immersive virtual reality application. As described the mobile application allowed for navigation, selection and manipulation however in this application the focus will be on the navigation and selection interactions.

5.1 Virtual Environment

Since Unity 5 was also used to develop the virtual museum mobile application it was possible to reuse the virtual scene (Figure 5.1) created during the development of the mobile application (Chapter 3). The user wearing an Oculus Rift Head-Mounted Display can look around the room by head rotation. The virtual museum room and the other 3D models used were created by Sérgio Eliseu.

Figure 5.1: Virtual Museum room

5.2 Configurable Content

The system will load at the start up the (XML) configuration files. In this application those configuration files define what videos will be played on the TV monitor (see configuration file in Listing 5.1) and the content associated with each portrait: name, biography and portrait picture (see configuration Listing in 5.2). This functionality allows for anyone to edit the content presented in the virtual museum without needing any kind of expertise on the application development.

Sample XML configuration files for the Virtual Museum

- Movies: list of the file names of the movies that will play on the television inside the museum

```xml
<data>
  <videos>
    <video>Aveiro1</movie>
    <video>Aveiro2</movie>
  </videos>
</data>
```

Listing 5.1: Virtual Museum XML configuration file

Portrait data

- Name: Name of the personality
- Description: Short biography
- Image: Name file of the portrait image of the personality
- Transform: rotation and position of the portrait in the Virtual Museum.

```xml
<data>
  <person>
    <name>Manuel Firmino</name>
    <description>Manuel Firmino de Almeida Maia nasceu
    a 19 de Janeiro de 1824 em Aveiro.</description>
    <image>manuel_firmino</image>
    <transform xRotate="0.0" yRotate="0.0" zRotate="0.0"
    xTranslate="0.0" yTranslate="0.0" zTranslate="0.0"></transform>
  </person>
</data>
```

Listing 5.2: Portrait XML configuration file

5.3 Virtual Museum Features

Users can move around freely using either a controller-based or tablet-based interaction. The navigation techniques were previously described in section 4.2.3. The portraits are the main focus of interaction. In the tablet-based method the user can perform a selection by first pointing at any of the portraits (the tablet screen will display the portrait and name of the personality selected) and then press the select button (Figure 5.2).

Figure 5.2: Selection of a portrait in the tablet-based interaction.

After a selection the tablet will show a menu with the set of available options to select. Selecting the **Biography menu option** as the name suggests will show a short biography of the Aveiro personality associated with the selected portrait. The **Location menu option** works in a similar way to Location Viewer described in section 3.4 from the previous chapter. The user will see a panorama (when available) of the real world location where the statue of the selected Aveiro personality is located.

Figure 5.3: Menu displayed after the selection of a portrait in the tablet-based interaction.

The **3D Model menu option** will show a 3D Model of statue of the selected portrait. The

3D Model will be displayed on the top of the virtual tablet (Figure 5.4). The user can rotate the tablet and bring it close to view the statue with more detail.

Figure 5.4: View of the 3D Model in the tablet-based interaction.

Augmented Virtual Reality

In the Virtual Museum on Tablet application there was an Augmented Reality (AR) option where the user was able to see a 3D Model of a personalty by pointing the tablet camera to an AR marker. This is a typical usage of augmented reality. An AR marker is registered and associated with a 3D Model so when the device camera detects the marker the model will be displayed on the screen. Even though we are now working in a fully virtual reality system, we can still simulate the described augmented reality behavior. We can point the virtual tablet to a marker just as we would do in a real world interaction and then show the associated 3D Model in the virtual tablet screen. The concept is exactly the same and although everything is now technically virtual we are enhancing the user's current perception of reality that is augmenting the virtual reality allowing for new types of interactions and applications.

Figure 5.5: Augmented Virtual Reality mode

Regarding the implementation it is important to explain that in Unity every Game Object has a Layer associated. Mostly Game Object when created are associated with the 'Default' layer but it is possible to change layers and even create new ones with any particular name. Cameras in Unity by default render every Game Object in the scene however we can change the culling mask options so the camera only shows Game Objects associated with specific Layers. So first we place a camera in the back of the virtual tablet where typically the camera in a real tablet would be. This camera culling mask is changed so it does not render any Game Object with the 'Hide' Layer. In this case the portraits on the wall will serve as AR markers. We place the 3D models on the top of each marker and change their Layers to 'Hide'. Now in the Main Camera which is associated specifically with what the user's view we change the culling mask so the main camera does not render any Game Object with the 'Hide' or 'ARView' Layer. In other words while navigating in the virtual museum the user will not see those 3D Models placed in front of the portraits. When the user selects the **Augmented Reality menu option**, the virtual tablet screen will show what the camera placed in front of the tablet is rendering. The Layer of the 3D Model associated with the marker of the selected personality is changed to 'ARView' and the user will be able to see that 3D Model on the top of the marker in the virtual tablet screen (Figure 5.5).

Chapter 6

Usability Studies

This chapter describes the two experiments designed to evaluate the usability of several aspects of the developed system. The Hand Representation experiment aimed to study the effect of the type of hand representation used in a tabled-based interaction. The results and conclusions from this first experiment where applied to the main system and used in the following experiment. The second experiment was designed to evaluate the performance of controller-based and tabled-based methods with the developed interactions (selection, navigation) included in the system.

6.1 Hand Representation Experiment

This experiment was mainly designed to verify if having a virtual representation of the user's hands would improve the menu selection task performance in a tablet-based interaction within an immersive virtual environment. It also aims to study a sub-question about whether the users would prefer a realistic or a transparent hand representation. In order to verify the effects of the three types of hand representation an experiment was designed where the users had to perform a set of button selection tasks for each type of hand representation. The next section describes the methodology followed by the analysis and discussion of the experimental results.

6.1.1 Methodology

Experimental Design

The experiment used a within-subjects design ([Dix et al., 2003]) where the participants had to perform a **Button selection task**. The task consisted in selecting as fast as possible the blue highlighted button from a group of 25 buttons that appeared on the virtual tablet screen. Before the task started, only one button appeared on the virtual tablet screen. The participant could use this part of the experiment to get used to the virtual environment and the equipment. Once this button was pressed the task started and the virtual tablet screen would display the set of 25 button being one of them blue. After the participant pressed the blue button another randomly chosen button was highlighted in blue. The subjects had to perform this button selection task 25 times for each type of hand representation, giving a total of 75 button selection tasks per subject. The **independent variable** was the type of virtual hand representation. The three conditions are described below:

1. **No virtual hand**: The user performed the button selection task while only viewing the virtual tablet. Every touch on the tablet touchscreen provided visual feedback consisting in a blue dot pointer on the position of the touch. Dragging the finger on the touchscreen would not be classified as selection (press button) only a tap touch would. In case the user pressed a wrong button, i.e. any button but the blue, that button would briefly turn red.

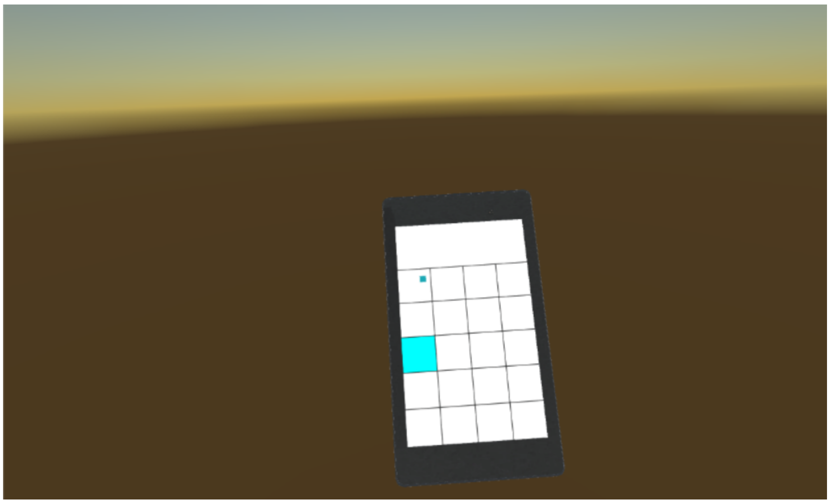

Figure 6.1: Virtual hand representation 1: No hands

---

2. **Realistic virtual hand**: In this condition besides what is described in the previous condition there is also a virtual representation of the user hands' movement. The 3D hand model is a realistic virtual hand representation that also includes the forearm and follows the real hands positioning and orientation captured by the Leap Motion mounted on the Oculus. The model used was available in the Leap Motion SDK for Unity 5.

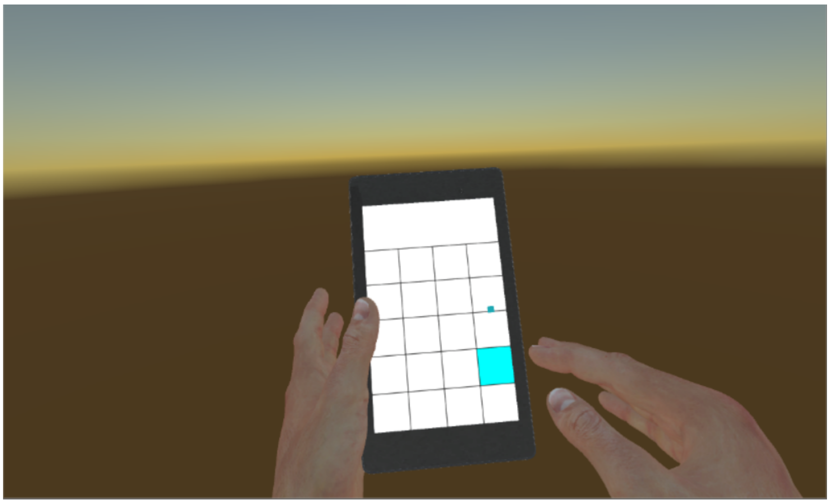

Figure 6.2: Virtual hand representation 2: Realistic hands

---

3. **Transparent virtual hand**: This third condition is very similar to the Realistic virtual hand but the hand model is now transparent in order to evaluate the effects of hand occlusion. The model is also from the Leap Motion SDK with a modified material.

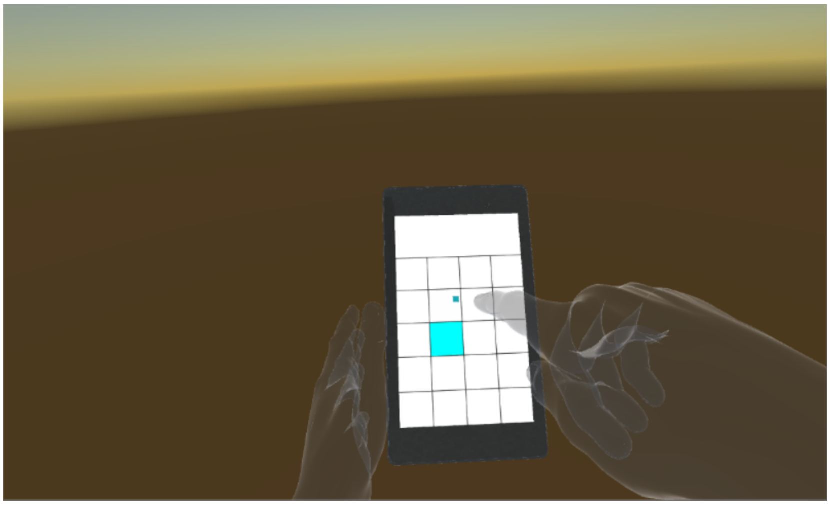

Figure 6.3: Virtual hand representation 3: Transparent hands

The dependent variables, automatically logged during the experiment, were the following:

1. **Task completion time (seconds)**: time taken to complete a button selection task. At the end of each block the average task completion time was computed as well as an overall time taken to complete the whole block (25 button selection tasks).

2. **Selection errors**: number of incorrect buttons pressed for each task. At the end of each block the average and total selection errors was computed.

The overall average time to complete the experiment was 2 minutes and 22 seconds per participant. The order of the hand type representation varied between participants. Since we had three different types of representation there were six possible order combinations that were sequentially attributed to the participants. This was done in order to avoid bias due to learning effects.

Apparatus and Participants

The main application developed on Unity run on a 15.6" laptop computer with a 1920x1080 resolution screen. The mobile device used as interaction device was a Google Nexus 7 tablet running the controller application (also developed in Unity). The IP addresses of the laptop and of the Nexus 7 were set in the controller application and main application respectively. The HMD, Oculus Rift DK2, attached to the laptop provided head tracking. A printed augmented reality marker was tapped to the front of the HMD so while the tablet front-camera could track the mobile device positioning and orientation was tracked. Depending on the type of hand representation the Leap Motion sensor was also attached to the computer via USB and mounted on the Oculus Rift DK2 for the hand movement tracking.

A total of 55 subjects participated in the experiment and 52 of those individuals completed the questionnaire. The 48 male and 4 female participants (aged from 19 to 28 years) were students from the Electronics, Telecommunications and Informatics Department (DETI) of the University of Aveiro. Most of the participants (40 from 52) use a smartphone or tablet device regularly and 30 of the subjects had never used virtual reality before.

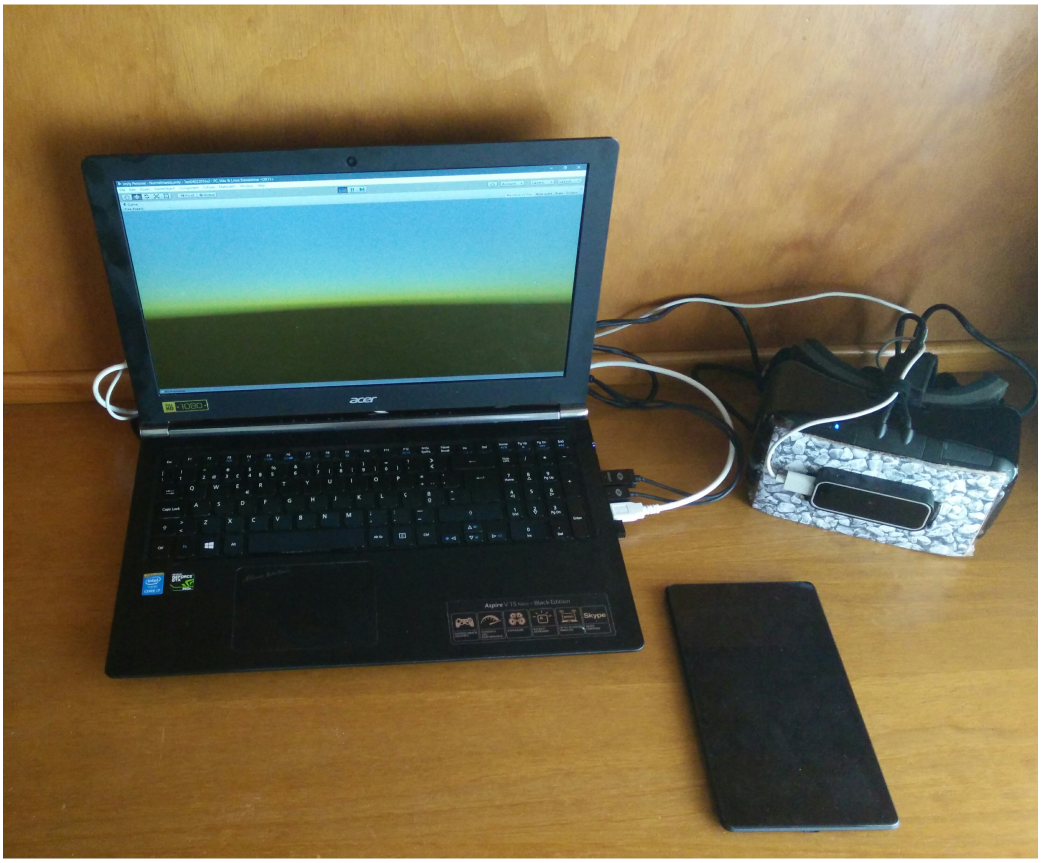

Figure 6.4: Experimental setup: Laptop computer running the Unity application; Nexus 7 Tablet; Oculus Rift DK2; Leap Motion Sensor (mounted on the Oculus Rift)

Procedure

At the beginning of the experiment, the subject was briefed about the equipment and the experimental task. The experiment was designed to be completed without interruption but the participant could remove the HMD if needed at the end of the completion of the task for each type of hand representation block. The final stage of the experiment comprised the answering an online questionnaire (Appendix A.1) with general questions about age, gender and if they had experience with the devices used and also questions related to each virtual hand representation (based on [Zielinski et al., 2014] questionnaire). Each user had an ID associated. The first three digits of the user ID were used to identify the order of hand representation. The no hand representation was number 1, realistic hand number 2 and transparent hands number 3. The last digit(s) was/were used to identify the number of the user in the experiment. This number increased sequentially along the users. For example a user with ID 21310 performed the button selection tasks first with realistic hands then with no hand representation and finally with transparent hands and it was the tenth participant in the experiment.

6.1.2 Results

As mentioned on section 6.1.1 (Methodology) during this experiment the task completion time and number of selection errors were logged. Figure 6.5 shows the time taken to complete the task considering each of the virtual hand representation types. From the box and whiskers plot, it can be seen that the subjects completed the button selection task faster when no virtual hand representation was shown (35.78 seconds average time) in comparison with Realistic Virtual Hand (44.42 seconds average time) and Transparent Virtual Hand (44.84 seconds average time).

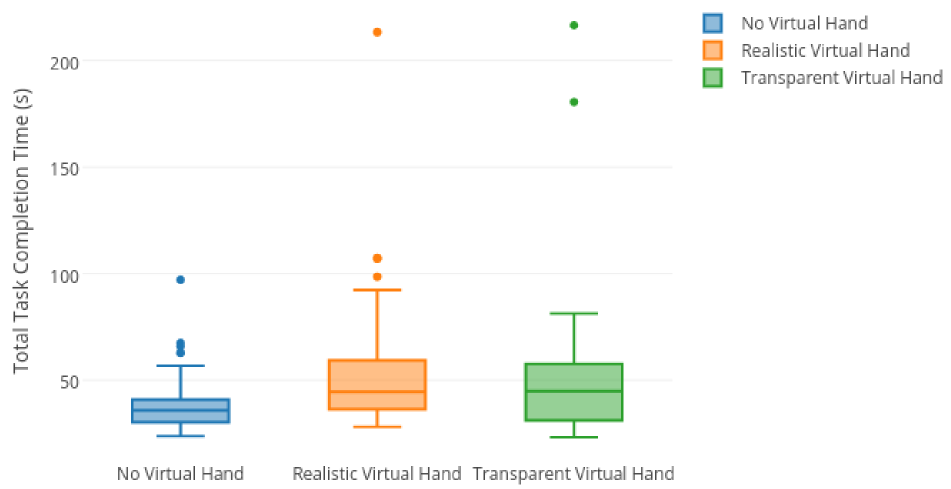

Figure 6.5: Box and whiskers plot of the task completion time considering the virtual hand representation. The outliers are marked as individual dots.

Figure 6.6 presents the average of selection errors during the experiment considering also the virtual hand representation. The average number of selection errors was relatively low in all three hand representation types. With an average total of 14 errors in 25 button selection tasks when there was no hand representation, total average of 12 errors in 25 button selection tasks with realistic virtual hands and total average of 11 errors in 25 button selection tasks) with transparent virtual hands. The results show that when there was a virtual hand representation (Realistic or Transparent) the subjects made slightly less selection errors when compared to no virtual hand representation being the transparent virtual hand the type with the lowest average of selection errors.

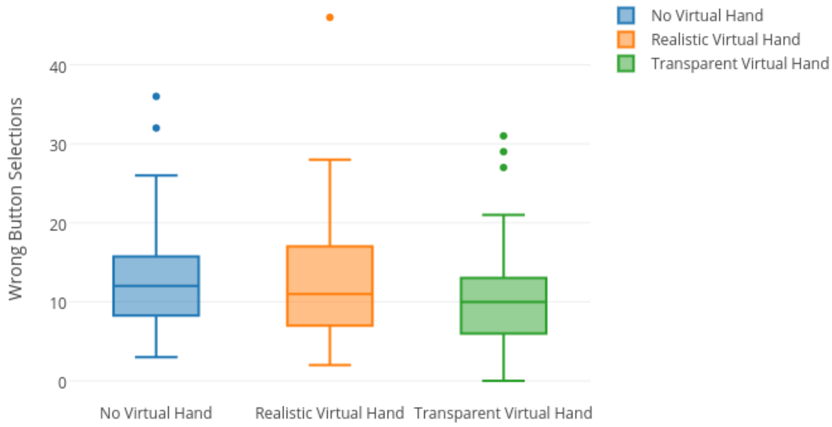

Figure 6.6: Box and whiskers plot of the total number of selection errors considering the virtual hand representation. The outliers are marked as individual dots.

Questionnaires

The results from the questionnaires were analyzed and are presented in Table 6.1. Only 52 from the total of 55 participants completed the questionnaire. When the participants were asked to rank the three virtual hand representation modes there was not a clear favorite. The respondents that indicated the No Hand mode has their favorite commented that they chose this mode over the other two because some of the (Realistic/Transparent) hand calibration issues that occurred during the experiment where the virtual hands position relatively to the tablet was inaccurate resulting in the virtual hands becoming a distraction to the user.

ID	Question	No Hand $\tilde{x}$	Reaslistic Hand $\tilde{x}$	Transparent Hand $\tilde{x}$
1	Please rank the three modes by preference	2	2	2
2	The task was (1 difficult, 5 easy) to perform	3.5	3	4
3	I felt like I was able to interact with the tablet the way I wanted to. (1 Strongly Disagree, 5 Strongly Agree)	3	3	3
4	I felt as if the virtual representation of the hand moved just like I wanted it to. (1 Strongly Disagree, 5 Strongly Agree)	NA	3	3.5

Table 6.1: Statistical summary for the questionnaire responses. The presented results are the median. For question 1 modes preference was ranked between 1 and 3 and the remaining questions were in a 5-Likert scale.

The results of question 2 show that subjects found the task easier to perform while using the transparent virtual hands. As one of the participant who preferred the transparent hand commented, the transparent hands give "the possibility of seeing the squares below the hand". Similarly, other users reported that the transparent virtual hand mode allowed to view the tablet screen and the hands without any obstruction. Regarding question 3, it can be seen that the majority were able to interact better with the tablet when the representation of the hands was transparent. In contrast, the participants felt less able to interact with the tablet while using the realistic hand representation. In response to the question 4, most of those surveyed tended to agree more with the sentence "I felt as if the virtual representation of the hand moved just like I wanted it to" in the transparent virtual hand representation.

6.1.3 Discussion

This study set out with the aim of assessing the importance of a virtual representation of the user's hands when using the tablet as an interaction device in an immersive virtual environment. Regarding the results of task completion time the subjects were slightly faster when no virtual hand representation was shown. A possible reason for the slower completion times in both the realistic and transparent hand representation may partly be explained by a degree of inaccuracy of the virtual hand positioning relatively to the tablet and tracking errors related to the Leap Motion sensor where the virtual hands would temporarily disappear or be upside down causing some distraction to the user. Improving the hand tracking accuracy would most likely improve the task completion time with the realistic and transparent hand representation and reach similar or even faster times to the ones obtained with no virtual hand representation If we now turn to the number of wrong selections results it is interesting to note that in all three virtual hand representation modes the subjects did not make many errors during the whole 25 button selection tasks. These results provide support for the idea of how natural is to interact with a tablet because how used to we are now to those kinds of devices even when inside an immersive virtual environment. It was not surprising to find that by having either the realistic virtual hand or the transparent virtual hand representation users were slightly less prone to make selection errors. The results from the transparent virtual hand being where there was on average less selection errors are in accord with the results from the questionnaires where the participants found easier to complete the experiment with the transparent virtual hands as well as felt like they were able to better interact with the tablet the way they wanted to.
These data suggest that having a virtual hand representation does improve the user experience in a tablet-based interaction compared to no hand representation. Also in general, it seems that the transparent virtual hands are more beneficial in comparison with the realistic virtual hands, however extending this experiment with more participants might provide statistically significant results.

6.2 Selection and Navigation methods experiment

This second experiment was conducted to evaluate the effectiveness of the tablet as an interaction device in a virtual reality environment compared to controller-based interaction. The participants performed selection and navigation with both a controller (Wiimote) and a tablet device. The virtual environment was the virtual museum room already used during the developing of the main platform. As in the previous section what follows is a description of the methodology and analysis of the data gathered in this experiment.

6.2.1 Methodology

Experiment Design

The experiment followed a within-subjects design ([Dix et al., 2003]) and was composed of two phases: the selection phase and the navigation phase. In both phases the subject performed the required tasks with both the Wiimote and the Tablet.

1. **Selection phase**: During this phase the participant would be facing the wall of portraits in the virtual museum room. This selection phase was divided in two stages. The user first had to select the yellow highlighted portrait from a set of portraits on the wall. After that the menu would be displayed (Figure 6.7). In this second stage the task was to select the 'Biography' menu option followed by the 'Back' button that would return the user to the menu. This time the user had to press the 'Location' menu option. While using the Wiimote, the user needed to look at the portrait (gaze input) and then press the 'A' button on the controller to confirm the selection. With the tablet the user points the virtual tablet to select the desired portrait and pressed the 'Select' virtual button displayed on the tablet screen to confirm the selection.

Figure 6.7: Selection phase. On the left: Menu interaction when using the Wiimote. On the right: Menu interaction when using the tablet.

---

2. **Navigation phase**: In this phase the task was to navigate through a series of yellow markers (six in total) in the virtual museum room (Figure 6.8). The subject started the experiment at the beginning of the virtual museum room. The first marker was placed in front of the user initial position. When the user reached the marker another marker would appear until the user navigated through all the six markers. The markers were predefined and in the same location for all the participants. In the controller method the user could navigate using Wiimote D-pad buttons and gaze. While using the tablet device there were two ways to move around and in both ways a map of the museum room would be displayed on the top half (white area) of the virtual tablet screen (see second and third image in figure 6.8). This map displayed the location of the marker, represented by a yellow square) and the current location of the user in the room, represented by a red square. The first way to move with tablet was using the tablet screen as a virtual joystick. The second way was using a teleport mechanism where the user could select any location in a museum room to be teleported to by clicking on the mini-map area. In the teleport technique on the bottom half (grey area) of the screen is also possible to move around using the virtual joystick.

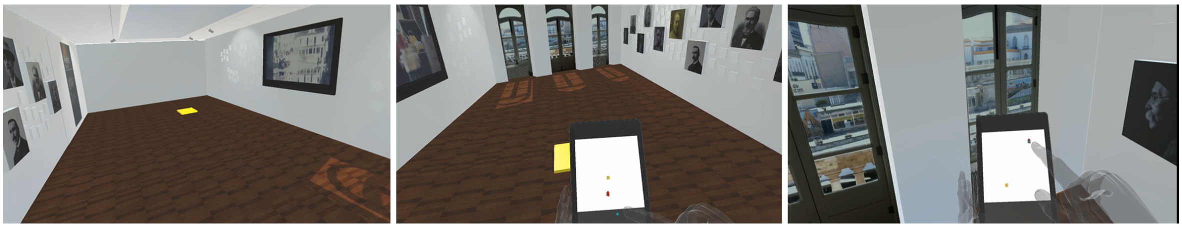

Figure 6.8: Navigation phase: On the left: Wiimote mode. On the center: Tablet mode. On the right: Tablet with Teleport mode

The independent variable was the interaction device that could be either the Wiimote (controller-based) or the Nexus 7 tablet (tablet-based). As in the previous experiment the dependent variables were automatically logged during process and were the following:

Selection

1. Portrait selection errors: number of errors when selecting a portrait.
2. Menu selection errors: number of errors when selecting a menu option.
3. Task completion time (seconds): time taken to perform the whole selection task.

Navigation

1. Marker time (seconds): time taken to reach each marker.
2. Task completion time (seconds): time taken to travel the specified route.

The overall average time to complete the experiment was 3 minutes and 48 seconds (1 minutes and 21 seconds for the selection phase and 2 minutes and 27 seconds for the navigation phase) per participant. The order of the hand type representation varied between participants. Since we had three different types of representation there were six possible order combinations that were sequentially attributed to the participants. This was done in order to avoid bias due to learning effects.

Apparatus and Participants

The setup was mostly similar to the previous experiment. The main application developed on Unity run on a 15.6" laptop computer with a 1920x1080 resolution. In the controller-based method a Wiimote was used as the interaction device. In the tablet-based method a mobile device (Google Nexus 7 tablet) served as the interaction device and was running the controller application (also developed in Unity). The IP addresses of the laptop and the Nexus 7 were set in the controller application and main application respectively. The HMD, Oculus Rift DK2, attached to the laptop provided the head tracking. A printed augmented reality marker was tapped to the front of the HMD so while the tablet front-camera was detecting this marker the mobile device positioning and orientation was tracked. A Leap Motion sensor also attached to the computer via USB and mounted on the Oculus Rift DK2 was used for the hand movement tracking (Figure 6.9).

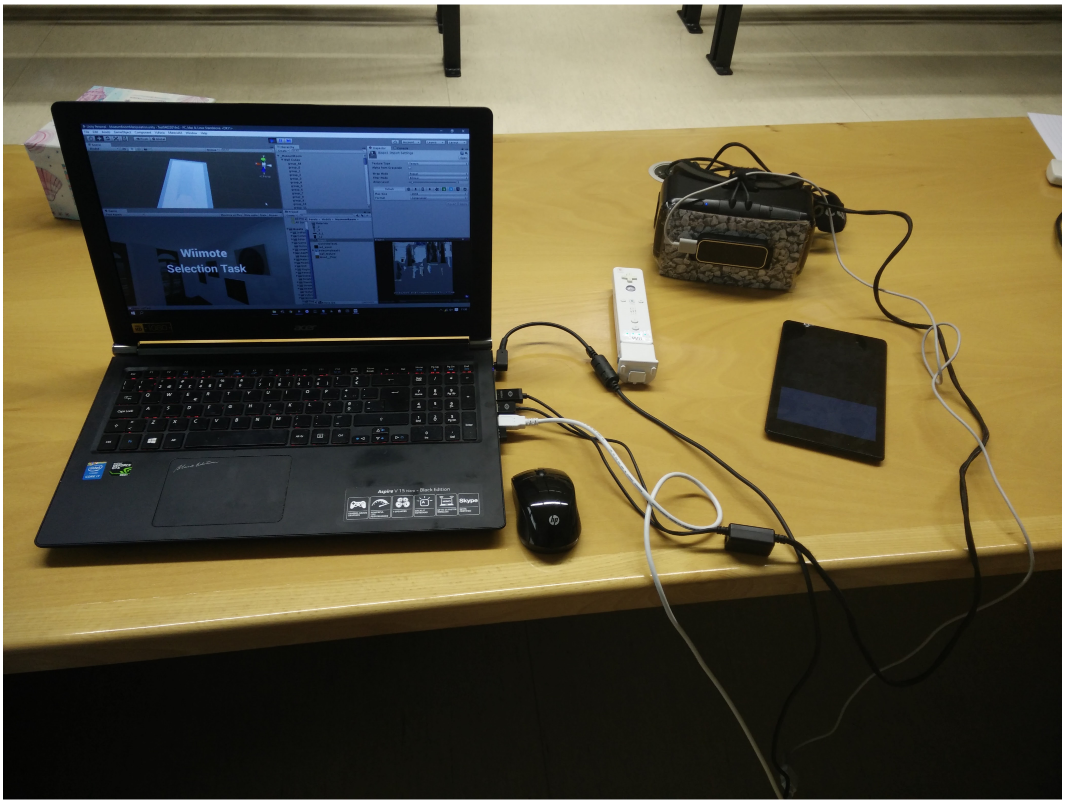

Figure 6.9: Experiment setup: Laptop computer; Wiimote; Nexus 7 Tablet; Oculus Rift DK2; Leap Motion Sensor (mounted on the Oculus Rift)

A total of 41 subjects participated in the experiment and 34 of those individuals completed the questionnaire. The 30 male and 4 female participants (aged from 19 to 24 years) were students from the Electronics, Telecommunications and Informatics Department (DETI) of the University of Aveiro. Most of the participants (26 from 34) use a smartphone or tablet device regularly and 11 of the subjects had never used virtual reality before.

Procedure

Prior to the experiment the participant was briefed about the equipment following by the explanation of the selection and navigation phases. The selection and navigation task were performed sequentially. As in the previous experiment (Section 6.1) each had an ID number associated. The first three digits of the user ID were used to identify the order of the interaction techniques. The Wiimote was number 1, Tablet number 2 and Tablet with Teleport number 3. In the selection phase, only the Wiimote and Tablet techniques were used. The last digit(s) was/were used to identify the number of the user in the experiment. This number increased sequentially along the users. For example a user with ID 12315, in the selection phase, performed task first with the Wiimote and then with the Tablet technique and performed the navigation task in the following order: Wiimote, Tablet and Tablet with Teleport. In the follow-up phase of the study, participants were asked to fill out an online questionnaire (Appendix A.2) related to the usage of the controller and tablet based methods while performing the navigation and selection tasks.

6.2.2 Results

During the Selection phase the time taken to complete the task and the number of wrong portrait and menu selection were logged. As shown in figure 6.10 the subjects completed the selection task faster with the Wiimote method (combination of Gaze Input and Wiimote button input) with an average task completion time of 27.1 seconds. When using the Tablet, the participants took on average 37.95 seconds to complete the task.

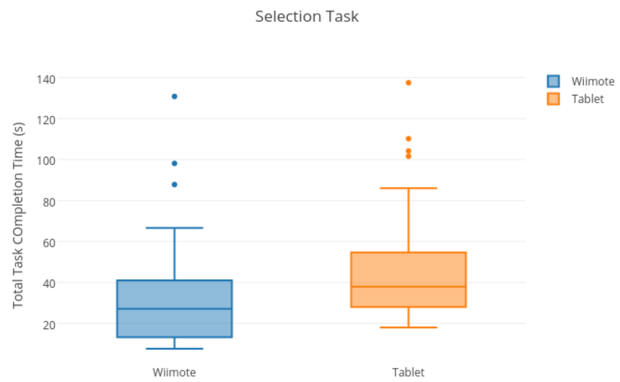

Figure 6.10: Box and whiskers plot of the selection task completion time considering the interaction device used. The outliers are marked as individual dots.

The first stage of the Selection task consisted in selecting the highlighted portrait. In figure 6.11 we can see that most of the subjects completed this stage without any selection errors in both conditions (Wiimote and Tablet).

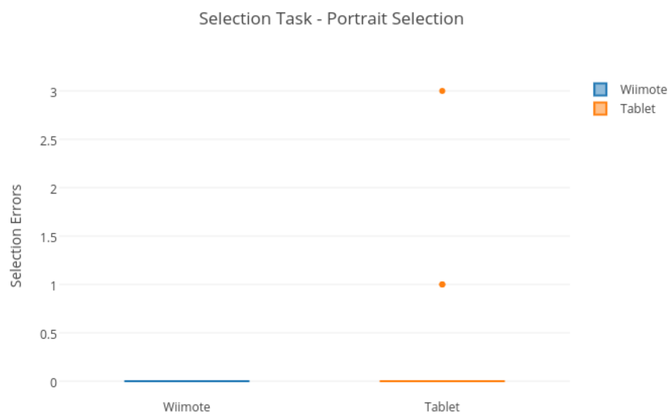

Figure 6.11: Box and whiskers plot of the total number of portrait selection errors considering the interaction device used. The outliers are marked as individual dots.

In the second and last stage of the Selection task the participants had to select a predefined sequence of button options as described earlier in section 6.2.1 (Methodology). Figure 6.12 shows the number of wrongly selected menu options. There were no significant differences between the two interaction devices.

Figure 6.12: Box and whiskers plot of the number of menu selection errors considering the interaction device used. The outliers are marked as individual dots.

After completing the Selection task the subjects would perform the Navigation task with 3 different methods: Wiimote, Tablet and Tablet with Teleport. The results obtained from the Navigation task are presented in figure 6.13 and it is clear that with the Wiimote the task completion time was on average lower than with the other two methods that used the Tablet. The Wiimote box plot is also comparatively short. What is interesting to note is that the lowest completion times for this task were significantly lower in both Tablet methods (23.3 seconds for the Tablet and 10.54 seconds for the Tablet with Teleport) in comparison with the Wiimote (31.26).

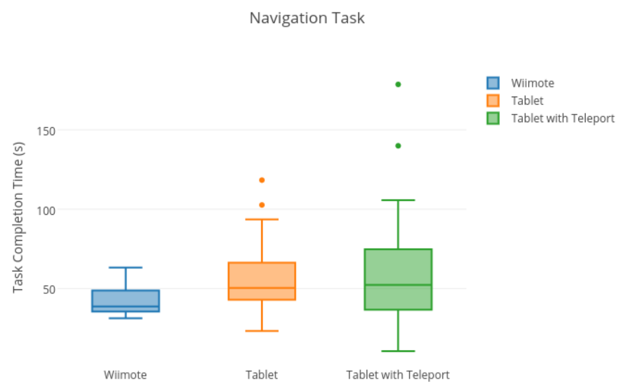

Figure 6.13: Box and whiskers plot of the task completion time considering the Navigation technique. The outliers are marked as individual dots.

The table below (Table 6.2) shows the average time taken to go to each marker for the three navigation techniques. What is interesting in this data is that from Marker 0 to Marker 1 there is a noticeable decrease in the difference between the Wiimote and the Tablet methods average times. In Marker 2, which had the longest distance relatively to the previous marker (Marker 1), the Tablet with Teleport method average time was fairly similar to the Wiimote method. At the end of the task, in the last marker (Marker 5) the difference of average times to the Wiimote is reduced to 21.43% for the Tablet (2) and 20.64% for the Tablet with Teleport (3). The Tablet-based interaction, (2) and (3), is still affected by tracking inaccuracies which causes some outliers. Since the average times are extremely affected by outliers we decided to also analyze the median times for a more robust statistic.

Marker ID	Distance to Marker	Wiimote (1) $(\bar{x} \pm \sigma)$ (s)	Tablet (2) $(\bar{x} \pm \sigma)$ (s) [Relative change to (1)]	Tablet with Teleport (3) $(\bar{x} \pm \sigma)$ (s) [Relative change to (1)]
0	6.45	3.97 ± 1.73	8.72 ± 8.69[117.88%]	9.73 ± 6.69 [145.09%]
1	11.40	5.26 ± 1.36	8.02 ± 4.08 [52.47%]	9.58 ± 7.73 [82.13%]
2	17.18	9.86 ± 2.78	11.89 ± 5.72 [20.59%]	10.28 ± 7.47 [4.26%]
3	15.53	7.04 ± 1.21	8.77 ± 2.63 [24.57%]	9.46 ± 7.01 [34.375%]
4	16.75	8.05 ± 2.21	9.06 ± 2.90 [12.55%]	10.59 ± 14.24 [31.55%]
5	16.08	7.56 ± 1.52	9.18 ± 2.67 [21.43%]	9.12 ± 7.42 [20.64%]

Table 6.2: Average times between each Marker considering the Navigation technique.

Table 6.3 shows the median time taken to go to each marker for the three navigation techniques. Figure 6.13 showed that subjects completed the navigation task faster when using the Wiimote but by analyzing table 6.3 we can see that there is a clear trend of decreasing difference between the times obtained with the Wiimote and the times obtained with the two Tablet navigation methods. As already observed for the average times (Tablet 6.2), from the initial Marker (0) to Marker 1 it is already possible to notice a prominent reduction of the difference between the median time of the Wiimote and the times of (2) and (3). This reduction continues in the following markers and in Marker 2 and 4 the Tablet with Teleport (3) was actually the fastest technique out of the three.

Marker ID	Distance to Marker	Wiimote (1) $\tilde{x}$ (s)	Tablet (2) $\tilde{x}$ (s) [Relative change to (1)]	Tablet with Teleport (3) $\tilde{x}$ (s) [Relative change to (1)]
0	6.45	3.5	6.88 [96.57%]	8.1 [131.43%]
1	11.40	4.82	7.14 [48.13%]	8.14 [68.88%]
2	17.18	8.96	11.96 [33.48%]	8.76 [-2.23%]
3	15.53	6.72	8.26 [22.92%]	7.8 [16.07%]
4	16.75	7.24	8.58 [18.51%]	6.7 [-7.46%]
5	16.08	7.12	8.66 [21.63%]	7.5 [5.34%]

Table 6.3: Median times between each Marker considering the Navigation technique.

Questionnaire

Table 6.4 shows the results from the questions associated with the Selection Task. There was a preference for the Wiimote to perform this task. According to the results of question 2 the participants have not found the Tablet method as easy as the Wiimote. As for the intuitiveness of both methods (Question 3), those surveyed felt that the Wiimote method was more intuitive. In Question 4 it is evident that subjects found the Tablet method required some training to get used to.

ID	Question	Wiimote $\tilde{x}$	Tablet $\tilde{x}$
1	Please rank the two modes by preference	1	1
2	It was easy to perform the selection task. (1 Strongly Disagree, 5 Strongly Agree)	5	3
3	This selection method is intuitive. (1 Strongly Disagree, 5 Strongly Agree)	5	4
4	This selection method requires training. (1 Strongly Disagree, 5 Strongly Agree)	2	3.5

Table 6.4: Statistical summary for the questionnaire responses for the Selection Task. The presented results are the median. For question 1 modes preference was ranked between 1 and 2 and the remaining questions were in a 5-Likert scale.

The results from the questions related to the Navigation Task are presented in table 6.5. Overall, the respondents did not have a definitive preferred mode (Question 1), however, the Tablet with Teleport was the mode that was less ranked as the least favorite (only 6 times) in comparison to the Wiimote (ranked by 14 participants as the least favorite mode) and the Tablet (ranked by 14 participants as the least favorite mode). When asked to justify the preference one of the participants said that although the Teleport mechanism requires some training to get used to it allows a more advanced user to move faster and also has the alternative of move with the virtual joystick. Similarly, another participant found that having both teleport and virtual joystick movement mechanisms quite useful.

ID	Question	Wiimote $\tilde{x}$	Tablet $\tilde{x}$	Tablet with Teleport $\tilde{x}$
1	Please rank the three modes by preference	2	2	2
2	It was easy to move to the desired position. (1 Strongly Disagree, 5 Strongly Agree)	5	4	4
3	This navigation method was intuitive. (1 Strongly Disagree, 5 Strongly Agree)	5	4	4
4	This navigation method requires training. (1 Strongly Disagree, 5 Strongly Agree)	1.5	3	3

Table 6.5: Statistical summary for the questionnaire responses for the Navigation Task. The presented results are median. For question 1 modes preference was ranked between 1 and 3 and the remaining questions were in a 5-Likert scale.

Regarding Question 2, those surveyed considered that it was easier to move to the desired position when using the Wiimote and in the last Question (4) we can observe that participants tend to agree that both Tablet navigation methods (Tablet and Tablet with Teleport) require more training in comparison to the Wiimote. As for Question 3, users felt that the Wiimote was the most intuitive navigation method. Nevertheless, the Tablet and Tablet with Teleport methods were still perceived as quite intuitive.

6.2.3 Discussion

This second experiment aim was to compare the Wiimote and the Tablet while performing Selection and Navigation Tasks in an Immersive Virtual Reality application. Starting by the Selection Task, the results from the selection task completion time (where the Wiimote was slightly faster than the Tablet) are in line with the questionnaire results where the subjects found the Wiimote easier to use in this task and generally agreed that the Tablet method required training. Although in the participant's opinion the Tablet was less intuitive than the Wiimote, both methods had overall a similar number of portrait and menu selection errors. The slower task completion times can be explained in part by the system still being prone to tablet and hand tracking inaccuracies. Improving the tracking component would probably provide a better use experience of the Tablet resulting in faster selection times. In general, therefore, it seems that Tablet is a usable method to select virtual objects and interact with menus and with improvement of the tracking it should be comparable to the Wiimote method in terms of speed.

In the Navigation Task the task completion times were shorter and overall more consistent while using the Wiimote than the other two methods. The results from Question 4 "This navigation method requires training" for the Tablet and Tablet with Teleport are consistent with the task completion time data obtained (Table 6.2 and Table 6.3). Although these Tablet methods require more training, their performance rapidly improves over time, even in a small experiment, and perform similarly to the Wiimote with the Tablet with Teleport surpassing in a couple of cases the times obtained with the game controller. It is interesting to see that even with Wiimote results being overall better than the other methods the participants still preferred the Tablet with Teleport with the Tablet method having a similar preference to the Wiimote. According to these data, we can infer that the Tablet is also a functional interaction device for navigation in an immersive virtual reality application. It can provide more interesting and faster ways for the users to move (Teleport and controlling the movement speed) than using controllers as the Wiimote. The ability to easily show information like a map with relevant locations of the virtual environment and without obstructing the view of the user is also a great advantage of the Tablet methods. The teleport mechanism was tested in this experiment in a small virtual room and already showed in some cases better results than the other methods. It is safe to assume that using the Tablet with Teleport method would be even more advantageous in larger virtual environments. As already mentioned, the tracking of the tablet and hands are still not perfectly accurate which causes some interaction delay when using the Tablet methods. This inaccuracy factor also causes sometimes the user to not be able to perfectly choose the desired position to teleport in the mini-map in the Tablet with Teleport method.

Chapter 7

Conclusion and Future Work

The main objective of this dissertation was to explore the use of smartphones and tablets as interaction devices in immersive Virtual Reality (VR) systems and how they compare to controller-based interaction. The literature on this subject showed already VR systems using mobiles devices to perform 3D interactions in virtual environments with results suggesting these devices present some advantages over 3D controllers. Another purpose of this work was to develop a virtual reality system where the virtual environment and the available virtual content were easily configurable.

A mobile application was first developed for a virtual museum application and provided a good virtual environment scenario to investigate some of the possible interaction styles approaches using the mobile device's touchscreen and onboard sensors to perform navigation, selection and manipulation interactions. The mobile application had two different modes to visualize and interact in the virtual museum the first only using the mobile device screen and the second using additionally a Virtual Reality head mount (Google Cardboard) to provider a more immersive VR experience. The development of this mobile application provided an important initial contribution to expand our knowledge on what these kind of devices (smartphones and tablets) can offer to improve the way we interact with virtual environments.

From these insights we started the development of a fully Immersive Virtual Reality system using both controller and tablet-based interaction. The system provided an easy way to configure some of the virtual content (videos, images and text information) through XML configuration files. The main focus of the development was the interaction using the tablet. A few ways to track the device were studied and implemented: 6DoF tracking device (WinTracker III), Augmented Reality (AR), Leap Motion sensor and Google Tango. The Augmented Reality was the selected method to track the tablet as it is simple to implement in any VR system since it does not require any additional hardware (besides the AR marker placed on the front of the HMD). It also offers a decent tracking capability, however, ways to improve it should be studied in future work. Another import aspect studied was the kinds of feedback provided in a tablet-based interaction. This issue led to a set up of one of the two usability studies conducted. The Hand Representation Experiment aimed to verify if having a virtual representation of the hand present would improve the interaction with the (virtual) tablet in an Immersive Virtual Environment (IVE) environment. The task required the user to select a series of buttons on the virtual tablet screen. This study used the Leap Motion sensor to track the hand and finger movement and AR to track the tablet and, besides the limitations of these tracking methods, the analysis of study results suggest that in fact having a virtual representation of the hands improves the interaction mainly by reducing the number of selection errors. There were two different virtual hand representation: a realistic and a transparent hand model. The majority of the users preferred the realistic virtual hand but the best results in terms of task completion and selection errors were obtained with the transparent virtual hands. This kind of virtual hand representation overcomes occlusion problems that not only occur with the use of the realistic virtual hands but when interacting with a mobile device in the real world.

To validate our IVE system the virtual museum mobile application developed previously was adapted into our platform. This application was then used as the main virtual environment for the second study where the performance in selection and navigation tasks was compared when using a controller-based (Wiimote) and a tablet-based interaction. The study suggests that a Tablet is a usable device to perform selection and navigation interactions in a Virtual Reality application. The results also suggest that using these kind of devices is rather intuitive, however initially they do require more training than controller-based method being after the first couple interactions somewhat equal or even better in terms of how quickly the action is performed. The mobile devices can easily present useful information about the virtual content in the application without reducing and obstructing the user's field of view.

This dissertation extends our knowledge concerning the interaction in Immersive Virtual Reality environments using mobile devices. It also provides an easily applicable new tracking method by using Augmented Reality. Moreover, the understanding of the effects of virtual embodiment in Virtual reality was enhanced.

Future work should focus on optimizing the tracking of the tablet and hand tracking to consequently improve the performance of the Tablet-based interactions. Future investigations should not only continue to explore the use of mobile devices to perform selection and navigation tasks in Virtual Reality environments but also focus on the manipulation interaction. The effects of having a virtual representation of the user's body in an immersive virtual environment on the immersion and interaction is also another area that needs further research.

Bibliography

J. Bauer, S. Thelen, and A. Ebert. Using smart phones for large-display interaction. In *User Science and Engineering (i-USEr), 2011 International Conference on*, pages 42–47, Nov 2011. doi: 10.1109/iUSEr.2011.6150533.

A. Benzina, M. Toennis, G. Klinker, and M. 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, pages 1519–1524, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0268-5. doi: 10.1145/1979742.1979801. URL `http://doi.acm.org/10.1145/1979742.1979801`.

L.-P. Bergé, G. Perelman, M. Raynal, C. Sanza, M. Serrano, M. Houry-Panchetti, R. Cabanac, and E. Dubois. Smartphone-based 3d navigation technique for use in a museum exhibit. *The Seventh International Conference on Advances in Computer-Human Interactions (ACHI 2014)*, pages 252–257, 2014.

D. A. Bowman. *Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application.* PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 1999.

D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004. ISBN 0201758679.

J. H. Chuah and B. Lok. Experiences in Using a Smartphone as a Virtual Reality Interaction Device. *Workshop on Off-The-Shelf Virtual Reality*, 2012. URL `http://joonchuah.com/wp-content/uploads/2012/09/chuah-otsvr-2012-smartphone.pdf`.

M. Deller and A. Ebert. *ModControl – Mobile Phones as a Versatile Interaction Device for Large Screen Applications*, pages 289–296. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-23771-3. doi: 10.1007/978-3-642-23771-3_22. URL `http://dx.doi.org/10.1007/978-3-642-23771-3_22`.

A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003. ISBN 0130461091.

S. Eliseu, P. B. Bastos, J. Pinto, P. Dias, and B. S. Santos. Museu Imaginário. *14 Encontro Internacional de Arte e Tecnologia: 14.ART: Arte e Desenvolvimento Humano*, pages 340–344, 2014.

B. Furht, editor. *Encyclopedia of Multimedia*, chapter Immersive Virtual Reality, pages 345–346. Springer US, Boston, MA, 2008. ISBN 978-0-387-78414-4. doi: 10.1007/978-0-387-78414-4_85. URL http://dx.doi.org/10.1007/978-0-387-78414-4_85.

H. Graf and K. Jung. The smartphone as a 3d input device. In *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, pages 254–257, Sept 2012. doi: 10.1109/ICCE-Berlin.2012.6336487.

J. G. Grandi, I. Berndt, H. G. Debarba, L. Nedel, and A. Maciel. Collaborative 3d manipulation using mobile phones. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 279–280, 2016. doi: 10.1109/3DUI.2016.7460079.

N. Katzakis and M. Hori. Mobile devices as multi-dof controllers. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 139–140, March 2010. doi: 10.1109/3DUI.2010.5444700.

S. L. Kim, H. J. Suk, J. H. Kang, J. M. Jung, T. H. Laine, and J. Westlin. Using unity 3d to facilitate mobile augmented reality game development. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 21–26, March 2014. doi: 10.1109/WF-IoT.2014.6803110.

D. M. Krum, T. Phan, L. C. Dukes, P. Wang, and M. Bolas. Tablet-based interaction panels for immersive environments. In *2014 IEEE Virtual Reality (VR)*, pages 91–92, March 2014. doi: 10.1109/VR.2014.6802066.

N. G. Lipari and C. W. Borst. Handymenu: Integrating menu selection into a multifunction smartphone-based vr controller. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 129–132, March 2015. doi: 10.1109/3DUI.2015.7131737.

D. Medeiros, F. Carvalho, L. Teixeira, P. Braz, A. Raposo, and I. Santos. Proposal and evaluation of a tablet-based tool for 3d virtual environments. *SBC Journal on 3D Interactive Systems*, 4(2):31, 2013. URL http://www.seer.ufrgs.br/jis/article/viewFile/41814/28438.

M. Nabiyouni, A. Saktheeswaran, D. A. Bowman, and A. Karanth. Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. In *2015 IEEE Virtual Reality (VR)*, pages 243–244, March 2015. doi: 10.1109/VR.2015.7223386.

J. Pinto. Virtual Reality Environment for immersive walkthroughs and interaction. Master's thesis, University of Aveiro, 2015.

B. Ries, V. Interrante, M. Kaeding, and L. Anderson. The effect of self-embodiment on distance perception in immersive virtual environments. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pages 167–170, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-951-7. doi: 10.1145/1450579.1450614. URL http://doi.acm.org/10.1145/1450579.1450614.

E. Saar. Touching reality using a touch device. Bachelor's thesis, Malmö högskola/Teknik och samhälle, 2014.

D. Souza. Plataforma para a configuração de ambientes virtuais interativos. Master's thesis, University of Aveiro, 2013.

A. Steed and S. Julier. Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 43–46, March 2013. doi: 10.1109/3DUI.2013.6550195.

A. Steed, Y. Pan, F. Zisch, and W. Steptoe. The impact of a self-avatar on cognitive load in immersive virtual reality. In *2016 IEEE Virtual Reality (VR)*, pages 67–76, 2016. ISBN 9781509008360.

T. M. Takala and M. Matveinen. Full body interaction in virtual reality with affordable hardware. In *2014 IEEE Virtual Reality (VR)*, pages 157–157, March 2014. doi: 10.1109/VR.2014.6802099.

T. M. Takala, M. Mkrinen, and P. Hmlinen. Immersive 3d modeling with blender and off-the-shelf hardware. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 191–192, March 2013. doi: 10.1109/3DUI.2013.6550243.

J. Wang and R. W. Lindeman. Object impersonation: Towards effective interaction in tablet- and hmd-based hybrid virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 111–118, March 2015. doi: 10.1109/VR.2015.7223332.

C. A. Wingrave, B. Williamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J. L. Jr. The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *IEEE Computer Graphics and Applications*, 30(2):71–85, March 2010. ISSN 0272-1716. doi: 10.1109/MCG.2009.109.

D. Zielinski, B. Macdonald, and R. Kopper. Comparative study of input devices for a vr mine simulation. In *2014 IEEE Virtual Reality (VR)*, pages 125–126, March 2014. doi: 10.1109/VR.2014.6802083.

Appendix A

Usability Studies Questionnaires

A.1 Hand representation experiment questionnaire

1. **User ID**: ____________________

2. **What is your age?** ________

3. **What is your gender?** ☐ Female ☐ Male

4. **Have you used Virtual Reality before?**

   ☐ Yes

   ☐ No

5. **Dominant hand:**

   ☐ Right

   ☐ Left

6. **How often do you use smartphone/tablet devices:**

   Never ☐—☐—☐—☐—☐ Regularly

7. **Please rank the three modes by preference:**
   No Hands (1) ________
   Realistic Hands (2) ________
   Transparent Hands (3) ________

8. **Explain why the mode [1/2/3] was your favorite:**

   ____

   ____

   ____

   ____

9. **How much physical fatigue did you experience in your arms while interacting with the environment?**

   None ☐—☐—☐—☐—☐ Extreme

No Hand Representation

10. The task was (1 difficult, 5 easy) to perform.

Difficult ☐—☐—☐—☐—☐ Easy

11. I felt like I was able to interact with the tablet the way I wanted to.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Realistic Hand Representation

12. The task was (1 difficult, 5 easy) to perform.

Difficult ☐—☐—☐—☐—☐ Easy

13. I felt like I was able to interact with the tablet the way I wanted to.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

14. I felt as if the virtual representation of the hand moved just like I wanted it to.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Transparent Hand Representation

15. The task was (1 difficult, 5 easy) to perform.

Difficult ☐—☐—☐—☐—☐ Easy

16. I felt like I was able to interact with the tablet the way I wanted to.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

17. I felt as if the virtual representation of the hand moved just like I wanted it to.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

18. Comments and/or suggestions about the equipment or the environment:

____

____

____

____

A.2 Selection and Navigation methods test experiment questionnaire

19. User ID: ____________________

20. What is your age? ________

21. What is your gender? ☐ Female ☐ Male

22. Have you used Virtual Reality before?

☐ Yes

☐ No

23. Dominant hand:

☐ Right

☐ Left

24. How often do you use smartphone/tablet devices:

Never ☐—☐—☐—☐—☐ Regularly

25. How much experience do you have with the Wiimote?

☐ Never used it

☐ Little experience

☐ Much experience

Selection

26. Please rank the two modes by preference:

Wiimote (1) ________

Tablet (2) ________

27. Explain why the mode [1/2] was your favorite:

____

____

____

____

Wiimote

28. It was easy to perform the selection task.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

29. This selection method is intuitive.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

30. This selection method requires training.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Tablet

31. It was easy to perform the selection task.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

32. This selection method is intuitive.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

33. This selection method requires training.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Navigation

34. Please rank the three modes by preference:
Wiimote (1) ________
Tablet (2) ________
Tablet with Teleport (3) ________

35. Explain why the mode [1/2/3] was your favorite:

____

____

____

____

Wiimote

36. It was easy to move to the desired position

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

37. This navigation method was intuitive.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

38. This navigation method requires training.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Tablet

39. It was easy to move to the desired position

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

40. This navigation method was intuitive.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

41. This navigation method requires training.

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

Tablet with Teleport

42. It was easy to move to the desired position

Strongly Disagree ☐—☐—☐—☐—☐ Strongly Agree

43. This navigation method was intuitive.

Strongly Disagree □—□—□—□—□ Strongly Agree

44. This navigation method requires training.

Strongly Disagree □—□—□—□—□ Strongly Agree