INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Augmented Reality and its aspects: a case study for heating systems.

Lucas Cavalcanti Viveiros

Dissertation presented to the School of Technology and Management of Bragança to obtain a Master's Degree in Information Systems. Under the double diploma course with the Federal Technological University of Paraná

Work oriented by:
Prof. Paulo Jorge Teixeira Matos
Prof. Jorge Aikes Junior

Bragança
2018-2019

Augmented Reality and its aspects: a case study for heating systems.

Lucas Cavalcanti Viveiros

Dissertation presented to the School of Technology and Management of Bragança to obtain a Master's Degree in Information Systems. Under the double diploma course with the Federal Technological University of Paraná

Work oriented by:
Prof. Paulo Jorge Teixeira Matos
Prof. Jorge Aikes Junior

Bragança
2018-2019

Dedication

I dedicate this work to my friends and my family, especially to my parents Tadeu José Viveiros and Vera Neide Cavalcanti, who have always supported me to continue my studies, despite the physical distance has been a demand factor from the beginning of the studies by the change of state and country.

Acknowledgment

First of all, I thank God for the opportunity. All the teachers who helped me throughout my journey. Especially, the mentors Paulo Matos and Jorge Aikes Junior, who not only provided the necessary support but also the opportunity to explore a recent area that is still under development. Moreover, the professors Paulo Leitão and Leonel Deusdado from CeDRI's laboratory for allowing me to make use of the HoloLens device from Microsoft.

Abstract

Thanks to the advances of technology in various domains, and the mixing between real and virtual worlds. Allowed this master's thesis to explore concepts related to virtual reality (VR), augmented reality (AR), mixed reality (MR), and extended reality (XR).

The development and comparison of Android applications and Microsoft HoloLens aimed to solve a deadlock in the recognition of instructions by the users. We used an interactive manual of assembly and disassembly for taps of residential heaters.

Therefore, this work deals with three main parts. Firstly, the exploration of the concepts of VR, AR, MR, and XR. Secondly, 3D modeling and animations techniques. Finally, the development of applications using Vuforia, Wikitude, and MRTK.

The users tried our application "HeaterGuideAR" to verify the effectiveness of the instruction passed by the interactive manual. Only a few users had some difficulties at the beginning of the trials. Thus, it was necessary to provide aid tools. However, other users were able to disassemble the faucet without any external help. We suggest continuing this work with more explorations, models, and situations.

Keywords: Augmented Reality, Mixed Reality, Extended Reality, 3D Modeling, Vuforia, Wikitude, MRTK, Interactive Manual.

Resumo

Graças aos últimos avanços tecnológicos em diversas áreas deram a possibilidade de fazer a mistura do mundo real com o virtual. É com este intuito que esta tese de mestrado veio expor os conceitos relacionados à realidade virtual (RV), realidade aumentada (RA), realidade mista (RM) e realidade estendida (RE).

O desenvolvimento e comparação de aplicativos Android e Microsoft HoloLens teve como objetivo resolver um impasse no entendimento de instruções por parte dos usuários. Utilizamos um manual interativo para montagem e desmontagem de torneiras de aquecedores residenciais.

Este trabalho, portanto, lida com três partes principais. Na primeira, a exploração dos conceitos de RV, RA, RM e RE. Na segunda, modelagem 3D e técnicas de animações. E por fim, o desenvolvimento de aplicações usando Vuforia, Wikitude e MRTK.

A aplicação "HeaterGuideAR" foi testada pelos usuários afim de verificar a eficácia da instrução passada pelo manual interativo. Apenas alguns usuários tiveram algumas dificuldades no início dos testes. Sendo que, foi necessário fornecer algumas ferramentas de auxílio. Mesmo assim, outros usuários conseguiram desmontar a torneira sem ajuda externa. Sugerimos continuar este trabalho com mais explorações, modelos e situações.

Palavras-chave: Realidade Aumentada, Realidade Mista, Modelagem 3D, Vuforia, Wikitude, MRTK, Manual Interativo.

Contents

List of Figures

Acronyms

API Application Programming Interface. 52

AR Augmented Reality. 1

CAD Computer Aided Design. 21

CAVE Virtual Automatic Cave Environment. 33

DOF Degrees Of Freedom. 20

GUI Graphical User Interface. 1

HMD Head Mounted Display. 13

HPU Holographic Processing Unit. 54

IoT Internet of Things. 8

JDK Java Development Kit. 52

MR Mixed Reality. 1, 42

MRC Mixed Reality Capture. 54

MRTK Mixed Reality Toolkit. 81

SDK Software Development Kit. 51

VR Virtual Reality. 1

Vuforia Vuforia SDK. 51

Wikitude Wikitude SDK. 78

XR Extended Reality. 44

Chapter 1

Introduction

1.1 Presentation

The interaction between man and machine has progressed a lot recently, by making the machines simpler to operate, interfaces have become faster, enjoyable and more efficient. As a result, it was also possible to improve the response of instruction passed to the machine, once the evolution of interfaces, along with the processing power of the computers and improvement of the GPUs, has arisen what today is called Virtual Reality (VR).

Virtual reality, which is heavily used for virtual games, is not limited to this link, it covers areas such as simulation of equipment, training of personnel, validation of planning and prototyping of products. Some areas in which this technology is used have a lot of impacts, because it allows the production costs, reduces the security risk and minimizes the objects needed for simulation. Thus, being practicable in the monetary scope of the company and safe for those who will use.

Currently, the use of Augmented Reality (AR) can be attributed as a consequence of evolution and progress that VR generated in problem-solving. Because of this, AR for not fully overlapping real reality and totally obscure the world behind the Graphical User Interface (GUI), brings more dynamism and innovation to the technological areas. As a consequence, the Mixed Reality (MR) emerges, an area that is still in development but that may be a continuation of progress that the solutions of the projects and applications with VR and AR have generated.

Among some highlighted reasons, this work is intended to explore the concepts of VR, AR, MR, and XR combined with the creation of three-dimensional virtual models and the development of an application for training and assembling of a heating system provided by the company TechWelf[1]. In other words, this project aims to solve the difficulty encountered by the follow-up of the installation rules with an instruction manual in AR, so that the user can be able to maintain the equipment without having to hire a third-party service.

1.2 Challenges

Here are the causes that may have appeal to the expected final result of the production of the AR training application, which requires knowledge of several areas. Some of which can be highlighted:

- Creating and Mapping 3D Objects;
- Integration of 3D models to the game engines (virtual worlds);
- Programming interactive objects;
- Identifying plans or markers for plotting virtual objects;
- The distinction of the target-marker object of other objects in the environment, whether by similarity or proximity;
- Interference or shadow quality present in ambient light;
- Handling of the mobile phone and operator's instructions.

[1] www.techwelf.com/

1.3 General Objective

Develop a cross-platform app (Android and iOS) with Augmented Reality for visualization of instructions of disassembling and assembling of a residential heating system.

1.4 Specific Objectives

- Model and integrate all 3D objects created for the AR development platform;
- Confirm the instruction passed by the application;
- Use of markers for plotting 3D objects with AR and thus proceeding with the instructions for disassembly the thermostatic valve;
- Prescribe the steps for assembling the new valve for the heating system;
- Evaluate the performance and instruction effectiveness passed through AR application.

1.5 Justification

Is not from today that advertising follows technology. With the advance of hardware [1] for long ten years, has favored it greatly so that customers could mature enough and expand the global market. The velocity which Nintendo's Pokemon GO (and their marketing emphasized in AR) is an example of its potentiality, was well received by the users and also had a profit of around 2 billion dollars[2].

Big companies such as Apple, Microsoft, and Google are also investing more and more in this sector that makes a fusion between real and virtual. Apple has announced the launch of the ARKit framework for IOS 11 in 2017 promising great experiences for those who are interested in the new technology. While Google continues to invest heavily in ARCore, a framework for Android, which promises to run native software for more than

[2] `https://bit.ly/2Va72Uk`

250 million smartphones[3]. Microsoft together with HoloLens[4] covers Mixed Reality and brings another concept of how the future might be.

With these technological advances, the number of API investments [2] for this branch has expanded, Wikitude and Vuforia are one of the examples that facilitate the development in AR, abstracting the calculations of the matrices necessary for plotting the three-dimensional virtual objects in the real plane. In addition, for being able to be linked to game engines like Unity3D and Unreal Engine, the followers of this new market are still growing. Furthermore, companies offer free versions of their services, so it is necessary to pay royalties only after the generation of profit. Thus, areas such as education, health, arts and entertainment, architecture and design, security and advertising, can benefit from this sector (Figure 1.1).

In addition, to the several areas mentioned above, AR's also present for training and validation of processes [3]. As the user has faster learning compared to the reading [4] and in an intuitive way (visual aid) in real-time [3]. Learning through the interactive manual can be a good solution to assist in cases of assembling and disassembling heating systems.

Because most training systems that use VR/AR applications require real-time processing, the hardware and software used may suffer in terms of performance depending on the technology that is chosen for construction. In this way, the computational evolution itself is a very participatory factor in this process. Whereas such technologies were created several decades ago [1], definitions ended up modernizing, due to recent factors such as the multiplicity of platforms and the feasibility of software capable of dealing with multisensory elements.

While multiprocessing capability was restricted only to large computers before, now it has been expanded to microcomputers such as mobile platforms (tablets and cell phones), the emergence of tangible interfaces [3] and multimedia learning. As the representation of tasks that lead to the use of similar problem-solving processes creates the formulation of a more consistent mental representation [4]. The use of visual examples creates a

[3] https://www.androidauthority.com/google-arcore-version-1-6-932783

[4] www.youtube.com/watch?v=hglZb5CWzNQ

cognitive adjustment that leads to more effective and efficient problem-solving. Therefore, it validates that when using an application in AR behind brings benefits of: using a recent technology, increasing users cognitive performance for problem-solving and consider the examples of applications of success[5].

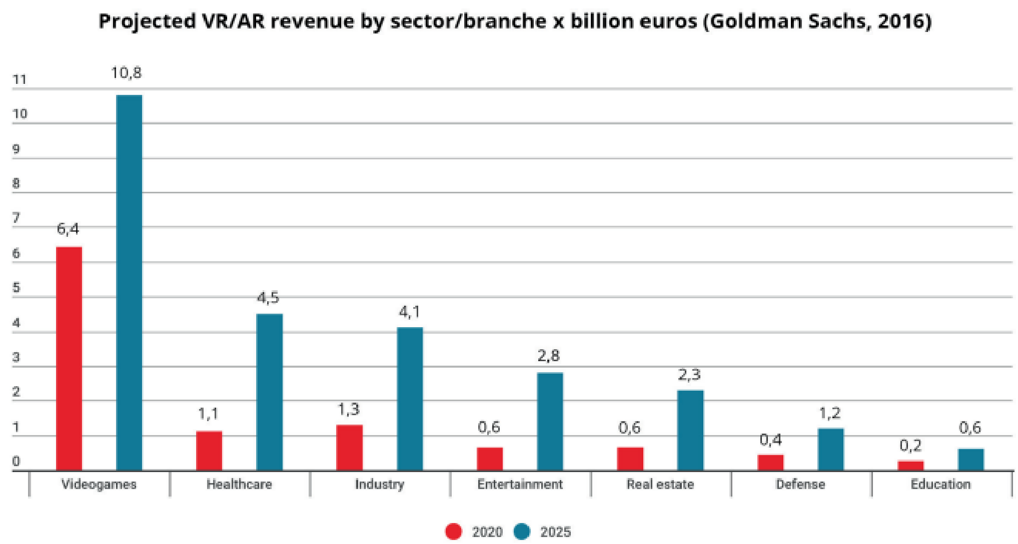


Figure 1.1: Growth Projection - VR/AR by sectors [2]

[5]BAE systems leverages the power of Mixed Reality: `https://www.youtube.com/watch?v=sDD-G32RqH8`

Chapter 2

Problem Analysis

2.1 The Heating Systems

In the current global context, the environment is one of the most important sectors that deserves a lot of care. As far as decision-making is concerned, it deserves attention, since the emission of gases can impact the increase of the greenhouse effect and, consequently, threatens the future of the planet and its inhabitants. The environmental impact of CO2 gases is too aggressive over the planet since the industrial revolution, there are still processes related to control the impact that these gases can cause in the ecosystem.

Air conditioning is one of the related areas, which directly or indirectly consumes the energy of resources that imply emissions of various pollutants. Therefore, the comfort that an air-conditioned environment provides, whether in cold or hot regions, can be increasingly associated with the environmental impact. Thus, it is unthinkable that for the vast majority of modern societies, using a central heating system, are not aware of the impact it will cause.

While in cold or temperate countries, the use of a system is attributed not only to comfort but also to a fundamental need for housing. These systems use water radiators and a boiler that makes the water heating so though that it can circulate on the environment. Radiators are air heat sinks that release heat to warm the environment. They also have a central control in the boiler that optimizes the operation, making the heating work in higher consumption regimes (higher temperature) only at specific times of the day, thus not only saving the energy but also reducing the gas emissions.

Today, with technological advances, Ultra-low Power and Internet of Things (IoT) have allowed the development of very efficient solutions that act locally at the level of each radiator. For the final consumer, to enjoy this utility is often summed up in a financial benefit (of replacing the control solution) in addition to savings that this generates.

Regarding the use of these technologies, there are two major components: the cost of control devices and their installation. Depending on the country, the second component may cost much more than the equipment. This makes many potential customers not choose or adopt these devices.

In this case, this master's thesis arose from a real challenge that aimed to facilitate the installation process of these devices, without the resources of specialized operators. The operation itself consists of the removal of the thermostatic head from the radiators and the subsequent placement of the new controller. It is not a complicated execution for those who have access to this kind of knowledge, but it is not yet available to the vast majority of users. Even so, there are several types of thermostatic parts and each one requires a certain type of instruction.

The challenge was then to create a way to replace the support of a specialized operator, allowing the user to do an auto-installation. The solution would make the placement of new controllers much more desirable. Thus, with natural advantages for the company that sells the devices, and also for users and, in general, for our planet. Consequently, the solution to be developed would necessarily have to be accessible to the common user, and interactively throughout the installation process, since this would facilitate the indications of how to carry out the steps in the process.

Therefore, according to the problems encountered by people who have a heating system in their home and have the need to perform the service for faulty faucet replacement, or even the heating system itself, there is a need to develop an alternative system, that takes into consideration the difficulties of the clients for removal and that also represents a new tendency of the consumer market. Moreover, it promotes the breakdown of the current paradigms: the requirement of using an outsourced service for the maintenance of residential heating systems.

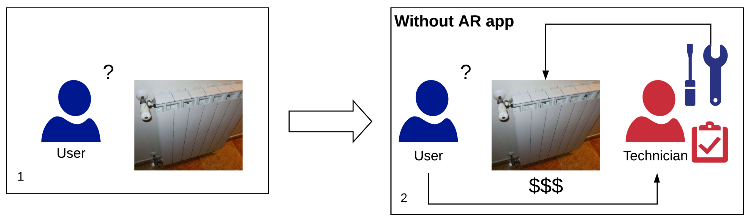


Figure 2.1: Outsourced service nowadays

Besides, some consideration should be given to the requirements and a more in-depth analysis of how the disassembling steps to occur. For example, to change the faucet, the holder or even the radiator, the heating circuit must be emptied first and when the heating circuit is solved, refill it in the correct pressure. If you need to replace a damaged radiator with a new radiator, you only need to close the faucet and the detent and disassemble it without having to empty the entire circuit. This is just one example of a specific system maintenance process. Another element that must be considered is that every thermostatic radiator faucet (to be fitted) must be positioned horizontally, more than 80 mm from the radiator, since the heat emitted by the radiator can disturb or damage the faucet [5]. Some points in question can be highlighted for the final solution of the training system because they can impact the performance and the use by the users. This chapter, therefore, aims to explore more about the problem that the present work aims to solve.

2.1.1 Types of heating system

The existing heating systems may have more than one type of technology for heating and temperature regulation. Both technologies must ensure the basic advantages of using a thermostatic faucet, therefore heating systems should:

- Modulate hot water flow automatically;

- Serve as an adjustment so that the ideal local temperature is maintained and, thus, avoid overheating or cooling the rooms of the house;
- Generate the temperature from the energy supply in calories of the central heating.

The heating systems also consist of: a valve equipped for manipulation, a bulb (sensitive to variations in ambient temperature) and a central heating radiator (Figure 2.2).

Figure 2.2: Radiator installed

Although Figure 2.2 shows few visible details, a heating system [5] can have three types of faucet head: cylindrical, mechanical and electronic. In the cylindrical head, the thermostatic faucet head consists of the hand-wheel and a bulb. The mechanical head, also non-programmable, is the most usual type of head. It can adjust the radiator heat automatically (from the position 0 to 5 or from 5 to 23 degrees). In this type, the body and the thermostatic head are sold together. Nonetheless, it is possible to buy the head separately and install it on a compatible thermostatic faucet body. Finally, the electronic head is the best solution that brings advantages in comparison with the other thermostatic heads:

- Has great precision and it is easily adjusted (up to half-degree);
- Can do hourly and/or weekly programming;
- Easy install (just need to insert the batteries in the head and fit it on the adapter previously attached to the radiator);
- The assembly can be done both horizontally and vertically (depends on the brand).

In addition to these points, the electronic head has an advantage over the others by providing reference temperatures, slots and profile of the rooms to be wired parameterized directly in the head or through a computer (via USB).

2.1.2 The adapters

Because there is a wide variety of brands and companies that operate in the heating system market, there is a wide variety of materials and respective products for this sector. Because of this, a problem arises regarding the lack of standardization of the adapters necessary for the installation of thermostatic taps. This implies that for the consumer, there is a difficulty determining the best adapter option and the compatibility between the new faucet model that the user wishes to install with the old one.

Besides that, some taps for being similar or with the same format of the model can be used by the same adapter. That is, one adapter can fit more than one model of thermostatic faucet.

However, because it is not a part of the users' prior knowledge, the users need instruction to help them, so that the application can better clarify how and which adapter to use according to the faucet that is to be uninstalled from their home, as well as being able to instruct them while removing and installing the adapters.

2.2 Solution

As the use of Augmented Reality in similar sector problems, as the proposed work, is increasing. It is desirable to use this new paradigm, since its use implies in the abstraction of the exposition ideas and, consequently, in the abstraction of technical instructions. That is, the use of an application turns the difficulty to understand an operation in a simple process from the forms of the 3D models visualized in real-time throughout the superimposed reality on the equipment of the user.

Some of the requirements that Augmented Reality and its aspects use were only possible thanks to the performance, the speed of computer processing and advances that the

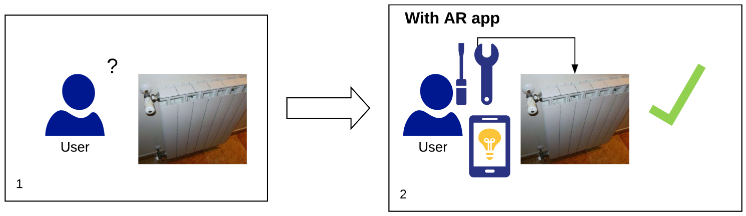


Figure 2.3: A user being guided through the app

current technologies hold. Thus, this work also aims at exploring, defining and acquiring knowledge and experience with the aspects that augmented reality currently has, and it is possible to make use of the latest and stable technology possible.

Therefore, it's necessary to understand the difference of associated areas and several concepts for creating the solutions, since it can also contribute to generating some basic knowledge for the community of developers, 3D modelers, engineers and other areas that can use the exploration and definition of segments that this work allowed to reproduce.

Chapter 3

Literature Review

3.1 A brief history of Virtual and Augmented Reality, Equipment and Advances

Through the temporal analysis of development and conception of Virtual Reality, one of the earliest traces found was not an application, but a tale: created by Stanley G. Weinbaum in 1935 "Pygmalion's Spectacles", was a science fiction that was based on glasses that allowed to make a holographic recording of fictional experiences, besides holograms generated in its lenses, allowed to perform the touch and the smell during the scene [6]. Although, the term VR was only defined in the late 1980s by the company VPL Research, whose name is associated with the artist, music composer and computer scientist Jaron Lanier [1].

While the denomination of the term Virtual Reality was only defined in the mid-1980s, prototypes such as the flight simulators built by the United States Air Force after World War II had already been developed in 1950 [7]. In 1961, the first prototype of Head Mounted Display (HMD) was developed by two engineers from Philco Corporation, Charles Comeau and James Bryan, Headsight was the first truly manufactured HMD similar to the current ones. The Headsight contained a magnetic motion tracking system to determine the direction of the head (Figure 3.1). This configuration is intended to be used in telepresence installations and remote viewing of dangerous situations by military personnel. The head movements then were associated with a remote camera, allowing the user to naturally look around and explore a remote location. Although it was something innovative for the time, the device did not integrate computer graphics and imaging.

Figure 3.1: Headsight [6]

Another prototype of VR was the Manhattan motorcycle ride simulator, patented by Morton Heilig in 1962 [7], the simulator with five short films allowed a multi-sensory experience with stereoscopic vision as well as sound and olfactory stimuli as seen in Figure 3.2.

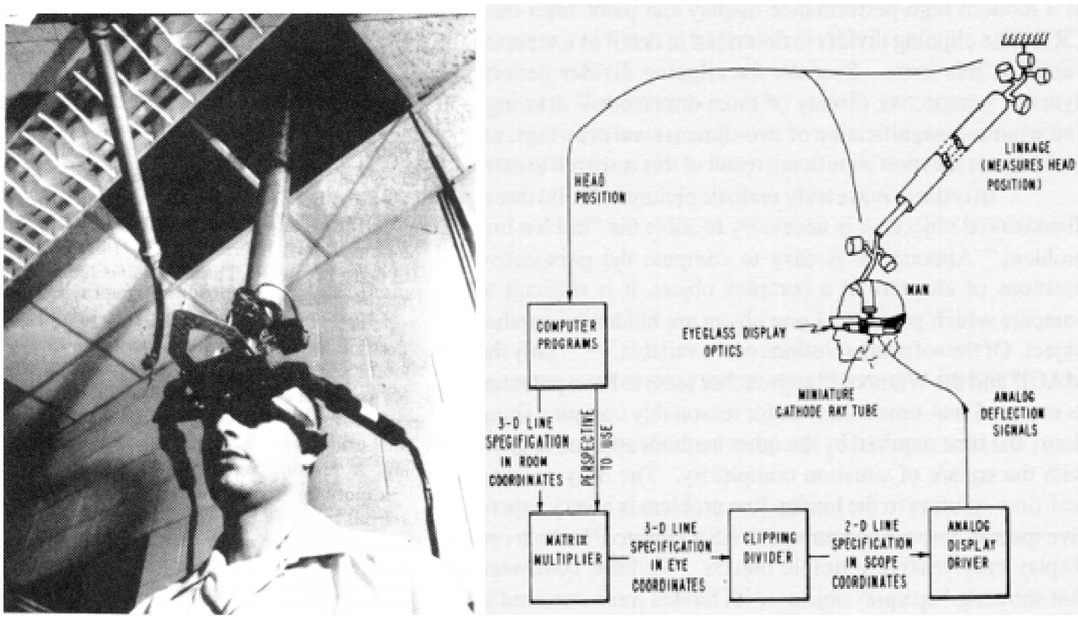
As early as 1965, the first optical pen was developed by Ivan Sutherland, which allowed for direct interaction between a real-world user and a computer-generated graphics through interactive virtual reality. Ivan Sutherland in 1968 also created the first HMD (Head Mounted Display) [9] system similar to the current ones, called the "Sword of Damocles" was a helmet-shaped apparatus similar to a periscope, which displayed the output of a computer program on stereoscopic 3D display. For being the pioneer, the industry was still very primitive in terms of graphical interface and realism, its graphics

Figure 3.2: Sensorama Morton Heilig [8]

comprised a virtual environment with simple rooms of 3D models of wire-frames, the user could then change the position and have different views [1]. It can also be said that it was the first display of Augmented Reality [10], the equipment needed to be suspended to the roof because the weight was too heavy to be used as can be seen in Figure 3.3.

Notably, another reality simulation project in virtual cities was the Aspen Movie Map, which in 1977 was created by Andrew Lippman to simulate a trip inside the city of Aspen, Colorado (USA) [11]. Considered as an interactive cinema for the epoch, this simulation was current based on the use of video discs that contained the photographs of the streets of the city of Aspen, so the films were obtained by cameras mounted in the cars that traveled through the city. The simulation could occur in summer, winter or through polygons mode, the first two being based on realistic photographs of the city and the last one through simple 3D models created.

In 1985, the term "virtual reality" was then, in fact, popularized by Jaron Lanier, one of the pioneers in the field of VR research and also the developer of some of the devices



(a) HMD (b) HMD Structure

Figure 3.3: HMD - Ivan Sutherland

that still continue to be exploited, such as the glasses and the virtual gloves. In the early 1990s, the interest in VR was huge, companies like W. Industries create the first VR games, the Ph.D. Jonathan Waldern demonstrates the "virtuality" of a staged exhibition in Alexandra Palace of London by means of Computer Graphics and in the same decade Sega announces the Sega VR headset for arcade games and Console Mega Drive [12], this headset used two LCD screens for display, the headphones stereo sensors in addition to the inertial sensors that allowed the system to track and react to user head movements. In the late 1990s, Virtual Boy was created by Nintendo and released in Japan and North America [12].

Starting in the 2000s, Google introduced Street View[1], a service that shows panoramic views of various points and locations around the world such as roads, buildings, streets, and rural areas could be seen on a computer screen. Although, the history of VR has shown progress in its development, from the first decade of the twenty-first century, the term "winter of Virtual Reality" has been attributed. Because there was so much media

[1] `https://www.google.com/intl/en/streetview/`

Figure 3.4: Aspen Movie Map [11]

interest in VR until mid-2012, only a few surveys have continued in corporate, academic, and military labs around the world [13].

In March 2014, Facebook acquires the company that manufactures virtual reality headphones and Oculus Rift [12], Google announces the Cardboard (stereoscopic viewer for smartphones). In the same year, Magic Leap became the focus of investors in AR/MR, including Google [14] itself. The Magic Leap One is an AR (and MR) device, that according to its creator, the technology allows generating images of the three-dimensional virtual objects indistinguishable from the real ones (Figure 3.5).

Figure 3.5: Magic Leap [14]

3.2 Virtual Reality

Despite the applications that this project aims to develop is in augmented reality, only with the advances of the virtual reality was it possible to conceive an AR, so this section aims to demonstrate some relevant themes of the virtual reality.

3.2.1 Describing virtual reality

Virtual Reality can be defined as an advanced user-interface, in which the user can navigate and interact through a computer simulated three-dimensional environment, being able to be completely or partially immersed into the sensation generated through the channels of sight, hearing, and touch [15]. Computer graphics, high-performance computing, and high-speed data transmission are the three major areas that involve VR. Some of the key elements that are also highlighted by the authors [16] are immersion (feeling of being within a virtual environment), interactivity (environment must react according to user interaction) and involvement (degree of user participation in a particular application).

Therefore, Virtual Reality can be characterized by the integration of these three basic concepts: immersion, interaction, and involvement. The immersion is related to the purpose of showing the user that when immersed in the virtual environment can have the experience or feeling of actually being inside an environment, even if it is being simulated.

In order to the user being immersed, it is necessary to use Input Devices (Section 3.3) that transmit the sensation of interaction in the virtualized environment (feeling of being inside an environment), making the connection of the sensory senses of the user and translating to the effect that happens inside the simulated space (interaction), with this intention makes the isolation to the outside world and allows to manipulate and to explore the objects instead of just being an observer (involvement).

3.2.2 Session types, degrees of freedom and navigation

Session types

The Virtual Reality session types can contribute to the degree of immersion, interaction and user involvement in the system, so according to the session's classification of the VR, there may be greater control of what will be done in a scene, generating a greater or lesser degree of freedom involvement and exploitation of the VR. A VR application can provide a session classified from three different perspectives: passive, exploratory or interactive session [17].

In the Passive session, the user makes use an exploration of an automatic and minimal environment without the interference of choice, the route and observation points of the simulation are explicit and controlled exclusively by the system itself, thus, the user has no control over how the scene occurs, except to leaves it.

In the Exploratory session, the user has more control over what will happen during the simulation, has a greater degree of freedom to explore the environment and can browse, search, cover more information and details of what will observe in the session (choice of routes and observation points), however there is no other way to interact with objects and entities belonging to the scene.

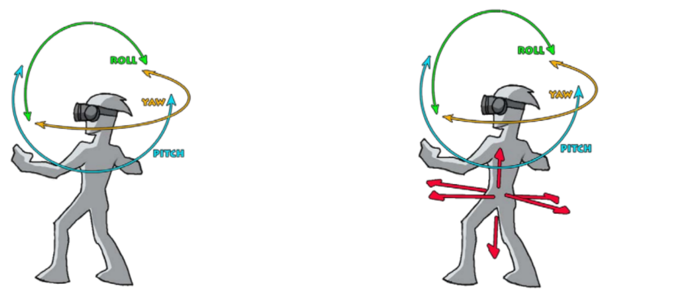
In the Interactive session, this provides a greater exploration of the user-driven environment and assigns feedback from the virtual entities of the environment, that is, the objects respond and react to the participant's actions. An example of interaction is when the user moves through an old wooden bridge, this one would have a swaying reaction as the user moves, giving the impression that it reacts as he/she walks.

Degrees of freedom and navigation

From the definition of the VR session types, it's then possible to define the navigations of the virtual environments that an immersive VR uses [16]. Thus, how the navigation in the environment can be controlled, such as the positioning of the observer in the virtual world and classification of the movements, are one of the key points to approach.

Developing an effective navigation technique for interactive 3D environments is an arduous task since there's a need to control the Degrees Of Freedom (DOF) [18]. In VR systems, DOF defines the users' ability to move around in space [19]. While in systems with 3 degrees of freedom - 3DOF (Figure 3.6a), allowed only angular vision control (rotation) in systems with 6 degrees of freedom - 6DOF (Figure3.6b) also covers positional placement (HTC Vive and Oculus Rift).

Therefore, the user can move forward or backward, move the view up, down, left and right and have the possibility to do slopes and rotations. Because of this, there are a large number of parameters to be controlled, besides the difficulties of control, there's the orientation problem, especially in large virtual worlds. An estimate of the Head Tracking should be made to be able to have a continuous angle measurement for the multiple DOF's [20].



(a) 3DOF (b) 6DOF

Figure 3.6: DOF

Interactive 3D environments usually represent more space that can be seen from a single point [18]. The user, therefore, needs to be able to move within the virtual environment, so that the user can get different views within the same scene. The navigation in virtual environments is generated from the positioning control of the observer in the virtual world, the viewpoint (or point of observation) when altered allows the user to have control of the visible perspective of the object or the virtual world which observes [17]. Throughout the analysis of the types of navigation, the ones that stand out are:

- **Point and fly** - The user moves the viewpoint (point of view) by pointing to a certain direction using an input device (glove or joystick, for example). Also called

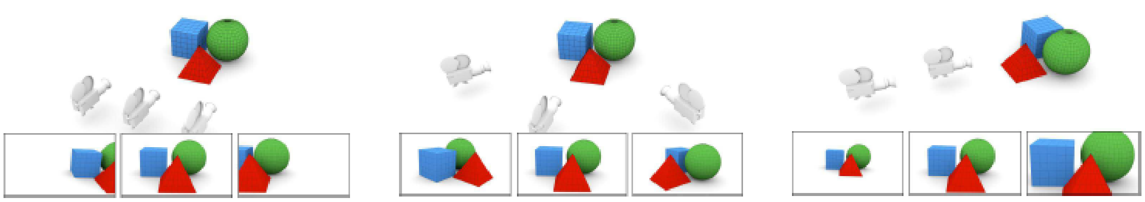
Directional Motion [18], the technique suggests that the user viewpoint changes to a specific object as the system computes the deviation in front of the object. The same technique is described as “Put-me-there travel” [21]. This navigation can be either in a simple way in which the user is taken to the specified location after a certain period, as in the real one or in an unnatural way, in which the user selects a destination and the system places it instantly in the place.

- **Eyeball in hand** - In this navigation model [22], the observation point remains fixed and the user can have different parameters from the same observation point from a single object, for example, a virtual museum in which allowed free space for movement around the objects. Thus, the user actions are updated according to the direction of the previously selected and focused observation point. The technique is also described as “Orbital-viewing” [21], in it, the user remains fixed and the objects of the world make the drive, as an opposite drive. In this technique the objects make an orbit around the user, depending on which direction the user chooses. For example, if the user changes the viewpoint to the left, then the field of view object will orbit to the right allowing the user to look to the left.
- **Scene in hand** - Similar to “Eyeball in Hand”. It makes an analogy to the "hands" of the user, from the mapping of the input movements, keeping the viewpoint of the user fixed and applying the move to the virtual world. In this technique, the user is able to have a mechanism of action of the hands in which he/she can pick up, drop, drag other objects, giving each of the movements. There are many parameters that affect the user representation and navigation, such as speed of the device, hand size, head movement range, and eye separation.

When creating VR and AR systems, you need to have control over certain parameters that can not be accessed by the camera or to the 3D objects focused (Section 3.3), Computer Aided Design (CAD) as Blender[2] and Maya[3], are examples that allow the following movements in a scene [18]:

[2] `https://www.blender.org/`
[3] `https://www.autodesk.com/products/maya/overview`

- General Movement (or Exploratory) - It is a basic technique of movement that uses three concepts for 3D interaction:
  - Pan - Refers to the act of translating the camera along the axes x and y according to Figure 3.7a;
  - Rotation - Refers to the act of orbiting the camera around a central point in any direction, both horizontally and vertically at the same reference point (Figure 3.7b);
  - Zoom - Refers to the approach or removal of the camera along its line of sight (Figure 3.7c)



(a) Translation (b) Rotation (c) Zoom

Figure 3.7: General Movement

- Specific Coordinate Motion - In addition to being used by developers of 3D Environments is used by Designers, it consists of moving the position and orientation of the environment. According to the view specified in relation to the model in the scene, an exact orientation and target position is provided. Thus, navigation can simply occur through data entry (for example on the keyboard) and thus carry out the change of coordinates in the x, y, z-axes.

3.2.3 Immersion types, sensory feedback, and requirements

Immersion types

Immersion is one of the relevant characteristics that help define a VR, according to the authors [16], there are two immersion categories: mental[4] (or psychological) and physical.

[4] https://www.youtube.com/watch?v=cXbtWX26VDw

As far as mental immersion is concerned, the authors cite that the user is so immersed that there is no suspicion of disbelief that that reality is being emulated. As a result, there is a great deal of involvement (for example: a child interacting with digital games). From the point of view of physical immersion, the user receives synthetic stimuli of the senses in the body through the use of technologies and this causes the body to appear immersed or partially (does not imply in all senses of the body) the simulation, that is, through sensory-motor resources create the sensation of physical contact and with it the sensation of presence, for example, a control that vibrates when making the point of observation hits some object of the scene of a game.

Both immersions have the same goal, to give the user the feeling of being inside an environment, to allow isolation from the outside world and to attribute the manipulation and exploitation of objects, but being more than just an observer in the virtual world. By making the users able to interact in real time, the situation change according to their command. This changes the users' perception and creates a sensation of fun. [23].

Another aspect of VR explored is VR without immersion which consists basically of the feeling of non-inclusion of experience in the virtual environment generated by a computer. What can be considered a non-immersive environment: visualization of three-dimensional images through a monitor, visualization of some game or in which the user interacts with the elements of the virtual environment through devices such as a mouse, keyboard, monitor and joystick. The advantage of non-immersion is given by the monetary cost since for immersion to happen it is necessary to invest in other Input and Output Devices like stereo VR glasses (Section 3.5.2), yet this "low-cost VR" makes it more accessible to the various user classes [21].

Sensory feedback

In order for the user to be immersed in a VR System, it is necessary to assign the Sensory Feedback to the physical position of the simulated world. Consequently, the computer must be capable of high-speed processing of the input data, in order to have the accurate sensory output of the system in VR.

According to the position of the participant in the world, the system must track the movements (Tracking) and give precise feedback, in the time the action occurs. Therefore, for a VR system to have control of the interaction, it is necessary to track some movements such as the user's head, hands or some object as a joystick or other input devices.

The VR is a virtualization of a computer simulated reality in which the environment, the objects and some of the characteristics of the objects and their present behaviors are also restricted and respect the performance of some law or natural physical behavior similar to real [21], gravity, for example.

Requirements

As explored in the previous sessions, it can be done a synthesis of the VR requirements. These requirements can impact the way the user is immersed, as well as how the user interacts with the simulation [17]. These requirements are:

- **Immersion**: the user must have the feeling of actually being inside a virtual world, regardless of how this immersion occurs;
- **Interaction**: the objects and the environment simulated by the VR system must react according to the actions of the user;
- **Involvement**: the system must assign ways of involving the user intuitively. Through the environment or situations that come from the common sense of the user such as: driving a car or controlling a virtual sword with the joystick;
- **High-quality Interface**: as the objective of a VR system is to give the user an interpretation of a similar reality, in which the user is accustomed. The VR interface must be the highest quality possible. Besides, the interaction between the user and the VR is merely intuitive so that the experience can be enjoyable.

From these points, it can be verified that a VR system can contain several attributes that when controlled prudently (processing and response of precise actions) can guarantee a better experience for the user. In a typical VR system, in addition to the basic definition requirements, it must also comply with certain physical hardware conditions [21] according to Figure 3.8. From tracing user actions in the virtual environment through the input devices, the simulation and response processing (graphics) are generated and verified on the output devices. Thus, a typical VR system should contain forms of tracking by Input Devices, high-quality rendering forms in generated graphics, and control of response to output data (visual and audible devices).

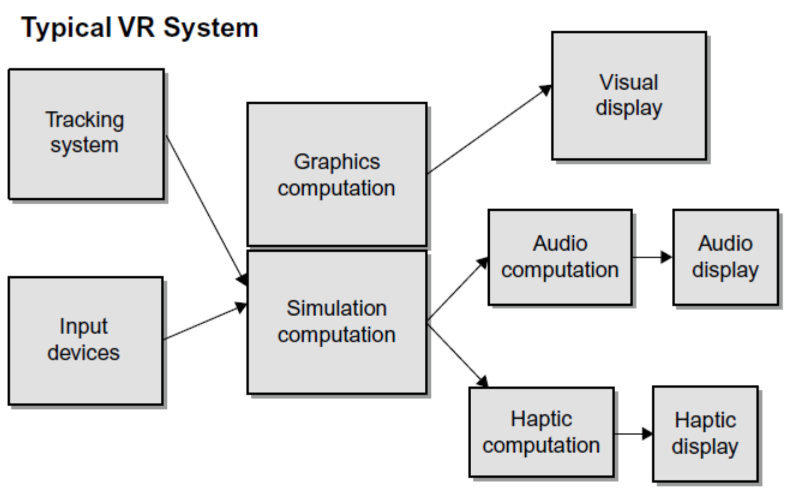


Figure 3.8: VR System [21]

3.2.4 Application examples

Although most VR systems and applications are intended for entertainment, there are a large number of projects focused on the field of "serious" applications. The following section presents some of the researches in progress.

Medicine

Through immersion that VR enables [16], users can have a presence experience in the simulated environment. Because of this, research and treatments for mental conditions (such as anxiety and specific phobias) can be explored in a virtual scene. These treatments are named as “virtual reality exposure therapy” (VRET) [24] and, for some cases, end up having quite advantageous results [25].

Among the applications of VR in the medical field, one of the most noted is the case of treatment of spider phobias [25], in addition for demonstrating the effectiveness of immersive VR, the research involved AR for the treatment and consisted of an assessment of user anxiety, avoidance, and behavior in relation to real spiders after the simulation of animated virtual spiders that were moving close to the user. The therapy was successful and managed to reduce the user’s fear of spiders by providing further evidence of the efficacy of using these technologies for exposure therapy.

Another example of VR applied to medicine was psychological therapy, which addresses the concept of user presence in relation to the attention of mental models visualized in virtual space. Research has shown that through virtual stimuli, these can impact the emotional phenomena of participants [26], and with this, the researchers were able to make a measurement of the study of what could affect the participant.

Analogous to the previous example, an exploration of VR in medicine consisted of assessing cognitive abilities by the system to deal with deficiencies of people who suffered brain damage. In addition to assisting in treatment, the research involved training strategies for rehabilitation of patients such as executive dysfunction, memory deficits, capacity deficits, attention deficit and etc. [27]. The review concluded that the use of VR in brain injury rehabilitation is expanding and will become an integral part of cognitive assessment and rehabilitation in the future.

Education and training

Educational and training systems may be part of the context of the exploration of VR [28] [29] [30]. Among the examples of education explored, the animation of the 3D anatomy of a computer-generated ear was a research that consisted of testing the educational effectiveness of an interactive model of the middle and inner ear through the magnetic resonance of the ear of a corpse. Medical students could use a VR to learn by an explanatory of the model based on the anatomy of the human ear [28]. The research concluded through questions that participants who had the experience would have more knowledge than the others who did not participate (This can emphasize the reason why using this kind of systems can help on the learning process as this work pretends).

In the same context, during a training aimed at surgical medicine was an AR system introduced to assist surgeons [29] in the practice of using real instruments, but with interaction with virtual objects. With the system, doctors could sense in real-time by the devices and make their surgeries simulated (the new Microsoft HoloLens 2 device is also conquering this sector).

In addition to the education areas, there are other VR systems applied to different risk situations [30] [31]. The Fire Simulator is a VR-based fire training simulator that takes participants to the risk of smoke. The technique consists of visualizing a situation of a building on fire, so they should learn and evaluate the safest way to get out of the risky situation, evacuate or rescue. Another training was the study conducted in the area of behavioral neuroscience, consisting of a weight experience in space environments [31]. The criteria for selecting indicators of performance and content control, as well as the relationship between sensor calibration and the influence of disorientation, proprioceptive illusions and changes in movement.

Culture and tourism

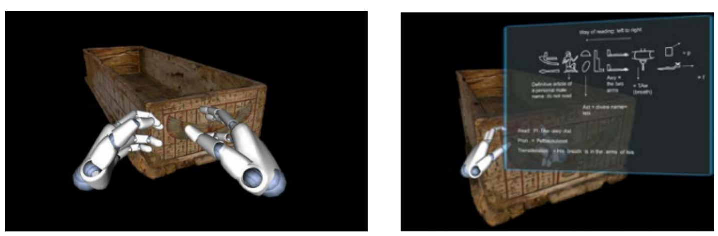
VR has great potential for exploring areas or environments of remote times [32] [33]. With regard to the exploitation of tourism, virtual museums, and the arts, there has been a recent trend [32]. The scene created by the education institution about immersive and interactive virtual archeology [32] demonstrates a representation of the Byzantine Empire (Figure 3.9a) and a representation of the Temple of Zeus (3.9b).

Another example is 3D models that can “preserve” historical artifact information. A research aimed at valuing and making more accessible the Egyptian funerary objects exhibited at Sforza Castle in Milan. The results of the research were used for the renovation of the current exhibition at the Archaeological Museum of Milan. An interactive 3D virtual scenario was created on the “Path of the Dead” in ancient Egypt, with which it

(a) Byzantine Empire [32] (b) Temple of Zeus [32]

Figure 3.9: Ancient Civilizations Tourism

was possible to increase the experience and understanding of the audience through interactivity. The developers used an Oculus Rift DK2, a Leap Motion controller and its implementation was done in Unity3D. All 3D models have been implemented and added to the points of interest responsive to important symbols or characteristics of the artifact. This made it possible to highlight isolated parts of the artifact in order to improve the identification of the hieroglyphs and provide their translation (Figures 3.10).



(a) Egyptian Scene 1 (b) Egyptian Scene 2

Figure 3.10: Egypt Tourism [33]

3.3 Input devices

As with the use of common operating systems, the input devices are devices that serve as an intermediate between the user's command and the receiving of data in the system. The fundamental task of input data is to move information from the user's brain to the computer. Thus, the human-computer communication channel can be established, from the real to the virtual.

Some researches involving input devices are concerned with the progress of useful bandwidth [34], that is, faster, natural, and convenient means of communication. While in the user's part, the input is limited by its perception, nature of the organs and senses [34]. On the computer side, it is restricted only by input devices which are the methods invented by technological advances to establish communication.

3.3.1 Keyboard and mouse

The most common devices for data input are the keyboard and mouse [34], joystick and trackball [35]. When using the keyboard and mouse, it allows to insert two-dimensional positions (2D) in the system, by mean of this, can fulfill the function of the input, when there is so much demand of interaction. In some cases [36], they still manage to be better than other newer input devices. Some of the effects of its use may be more effective in terms of speed and error rates of input [36] than other devices. Although VR systems require more and more peripherals, it is notable that the best technology employed depends on each VR system [35]. Just as current technologies are still in the process of adaptation, such as Visual Display Units (VDUs) that can cause physical, physiological, and psychological problems [35]. Therefore, the keyboard and mouse can be of good use for some VR systems, since they do not cause side effects.

3.3.2 Data gloves

Data gloves are input devices that allow you to recognize the movements of the user's hand. By recognition, the system must be able to determine which movement of the user's fingers are being triggered [17]. The user can then have a more common interaction with their perception and also visualize their actions in the VR system (feedback).

Although the use of mechanical sensors or fiber optics are the most used [17], there are several types of materials and technologies that can also be applied to the recognition of a data glove, among them, the most famous are [37] [38]:

- **Inertial**: is a standalone sensor that measures the rate of change of both the orientation and the variation speed of the movement of an object. The advantage is that by integrating "sourceless" technology it has no line-of-sight restrictions and low sensor noise;

- **Magnetic**: by means of a magnetic field generated by the tracker, it determines the position and the variation in the motion receiver. It has a reasonable precision and no requirement to the transceiver connection. The disadvantage is in the sensitivity to electromagnetic fields of other materials, which can generate signal interference;

- **Ultrasonic**: can determine the position of a moving receiver by means of the ultrasound signal that is generated by the stationary transmitter. Unlike magnetic scanners, ultrasonic scanners do not experience metallic interference. However, they may suffer from the transfer of information relative to the surface, which requires a direct line of sight from the transmitter to the receiver. If any object obstructs the line of sight between the transmitting ultrasound and the receiver, the signal is lost;

- **Optical**: through an optical sensor the tracker can determine the position and real-time orientation of an object. The advantage is that they are not subject to interference from metallic objects. However, it has significant upgrade rates. The disadvantages are seen in the sensitivity of the light reflection on the surfaces of the environment.

Examples of applications

Some of the technologies for haptic devices may help and encompass how to interact with VR systems. A model of haptic VR system can serve for a number of situations, in a specific case, it served as a basis for making feasibility tests on hand and arms training in people with cerebral semi-paralysis [39].

The project was based on the simulation of a virtual piano in VR and used a lightweight robotic peripheral, only 220 grams, for data entry called "CyberGloves". The user's actions on the simulated piano were then translated into the system and presented through visual, auditory and tactile feedback comparable to a real piano (Figure 3.11). According to the authors, "the ability to visualize a representation of one's own hand moving through virtual space may strengthen a participant's feeling of being involved in an action and of attributing that action to themselves" [39]. This emphasizes some of the topics discussed in previous chapters, such as the requirements of a VR system, immersion types, and sensory feedback.

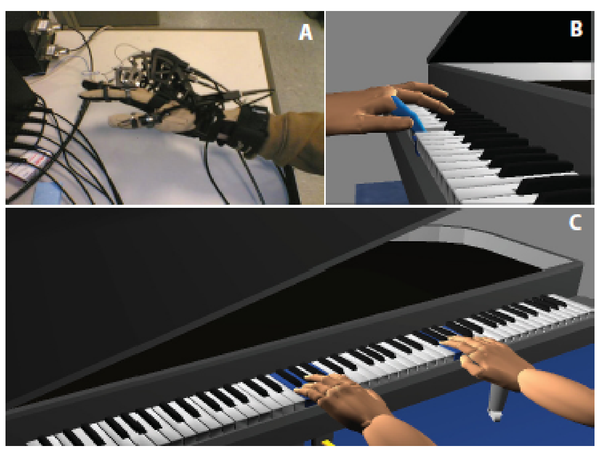


Figure 3.11: VR gloves [39]

Although data gloves are frequently used as devices in virtual reality environments, for recognition and selection of virtual objects. There are other manipulation techniques that require more than a simple haptic VR system available. Among such solutions are models capable of tracking the pitch, yaw, and roll of information from the user's hand, that is, the complete movement of the hands and fingers. With regard to full motion (x, y, and z-axes), tracking systems can have a higher financial cost than common haptics [40] systems. For this feasibility, one of the solutions that a project [40] aims to solve is to use markers for recognition and tracking of the user's hand according to Figure 3.12.

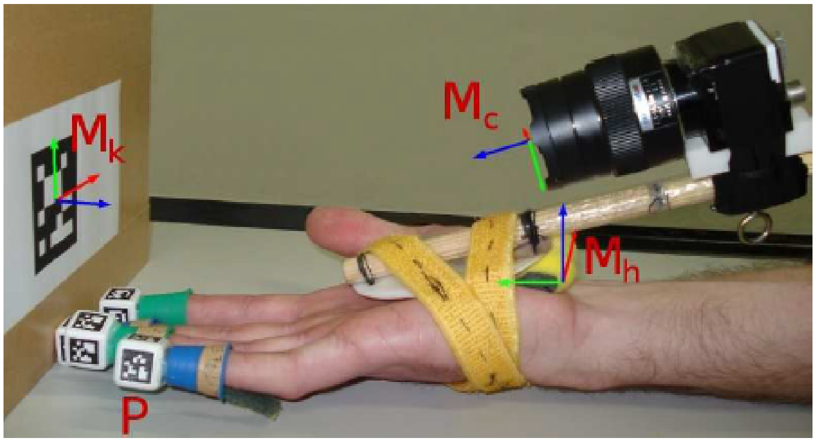


Figure 3.12: VR gloves [40]

Through a camera in the user's hand and visual markers for the fingers, the computer vision technique used consists of estimating the relative position of the fingertips. From the information, it is applied the technique of reverse kinematics that estimates the position of each joint and recreates the movements of the fingers in the virtual world. Thus, it is possible to map the pitch, yaw, and roll (x, y and z-axes) by recreating all the gestures performed.

3.4 Haptic devices

Also called tactile reaction devices [17], its basic function is to stimulate sensations such as touch, muscle tension, and temperature. Unlike common input and output devices, haptic devices require more sophisticated feedback and electromechanical interaction[5] with the body of the user.

In most commercial VR devices, there is tactile feedback [41], not only favoring immersion but also influencing user interactivity. The video game industry plays a key role

[5]Haptic glove - https://www.youtube.com/watch?v=OK2y4Z5IkZ0

in the cost of these devices [21], as they give control of user actions. Some joysticks that have vibration actuators (such as Xbox360, Nintendo Wii and PlayStation) are considered to be haptics devices [41].

Among the types of haptic feedback, the modalities of strength, tactile and proprioceptive feedback are grouped [42]. Force feedback is integrated into the simulation and provides data regarding the hardness, weight, and inertia of an object. The tactile feedback is used to give the user the feel of the geometry and smoothness, the slip and also the temperature in contact with the surface of the object. While, in proprioceptive, its feedback is generated by the perception of the position of the user's body, that is, its posture [42].

3.5 Output devices

The output devices are responsible for presenting the response of the virtual environment to the user. Through the output device, the user can see the processing of the response of the computer. Thus, these devices serve as an intermediary between the processing generated in the system and the receipt of the information by the user. Among output devices, there are categories of devices that fall into: visual, auditory, and haptic displays. [43]

3.5.1 CAVE

Developed at the Electronic Visualization Lab (EVL) at the University of Illinois, Chicago. The Virtual Automatic Cave Environment (CAVE), is an output device for visualization of virtual reality systems created in 1992 by Tom DeFanti and Dan Sandin. Before its creation, other forms of graphical representations, output devices, and integration methodologies were explored in the laboratory [9] [21].

Besides being able to represent different types of graphic images by means of a large, single screen [21], it gives the sensation of immersion and sensorial perception almost integral to the scene [17]. It also appears that the illusion of the 3D objects seem to coexist in the same space with users in a room [44].

By designing stereoscopic images on walls and having high-quality image resolution due to their lush size, CAVE's field of view is broader compared to HMD based systems [9]. Unlike other types of immersive environments, the virtual experience in CAVE is not limited to a single participant, multiple users can use, share and interact with the same information generated in the virtual environment [21].

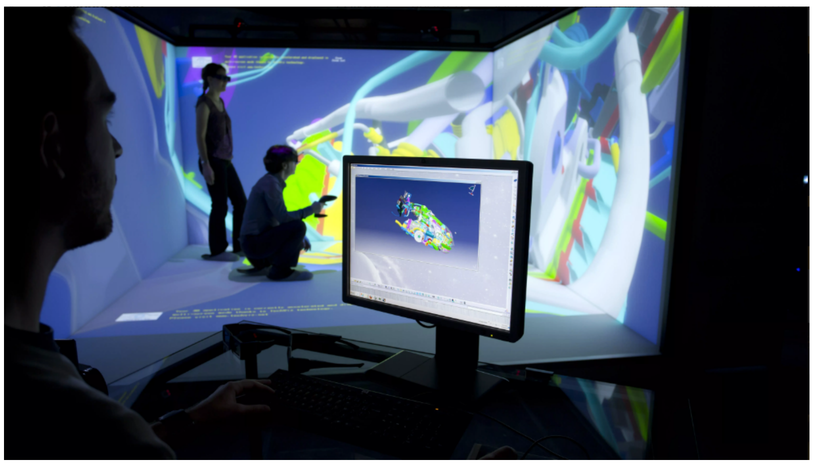


Figure 3.13: CAVE [45]

3.5.2 Visual Displays

HMD

The HMD or Head Mounted Display is one of the most popular devices associated with VR technology [17] (Figure 3.14). The immersion into the system takes place through a device which is formed by a set of optical lenses for viewing (LCD[6] or CRT[7]) and an imaging source. In the immersion experience with HMD, the imaging view is focused a few centimeters away from the user's eyes and the field of view ranges from 40 to 110 degrees.

[6]LCD - Liquid Crystal

[7]CRT - Cathode Ray Tube

[8]https://www.fastweb.it/var/storage_feeds/CMS/articoli/ba1/ba1869a1242f777b2aa91af9de0a5eba/Eyephone-VPL-1989.jpg

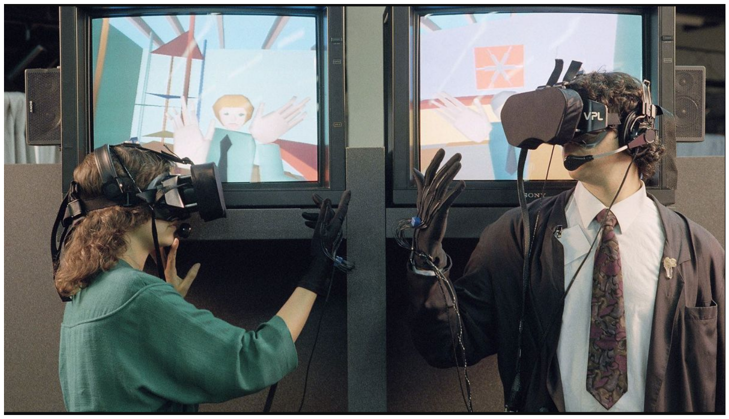


Figure 3.14: HMD[8]

3.6 Augmented Reality

3.6.1 Describing Augmented Reality

Augmented reality (AR) is related to the concept of Virtual Reality (VR) [46] [47] [48]. While VR tries to create an artificial world in which the user can have virtual experience and interactively exploration through his senses [46], augmented reality tries to assign the virtual into the real, in other words, its goal is to complement the real world. By making virtual object overlap the real world, it allows users seeing each other, as well as the space between them (real and virtual objects). Users can interact with the real world at the same time as virtual images are generated, bringing the benefits of real-world and VR interfaces together [47]. It can be said, then, that augmented reality is a consequence of evolution that virtual reality has allowed.

AR is present today, thanks to the opportunity to use some of the concepts of progress that VR had already been developing. The AR is described as the mixture between the real and the virtual [48], basically with the use of AR it is possible, with the current technological advances (in terms of computing power and quality of view), add information of the "virtual world" through the perception of the real world [49].

As far as interaction paradigms are concerned, AR encompasses a more natural and human familiarity, the physical environment and physical objects can exchange data with the virtual, allowing body language, movement, the look and physical consciousness itself trigger events that interact in virtual space.

In order to facilitate the perception of information, AR devices have been developing new concepts to improve the representation of information [48], although there is still a contrast regarding the handling of applications and equipment (lay users) that end up discourage its use. The concepts that characterize an AR system are, the clear and intuitive representation of the virtual information perception attributed to the real plane and the natural interaction with the user-generated interface [49].

3.6.2 AR System Architecture

In a typical AR system, it is necessary that its architecture has some key points [50] as ways to do tracking (corresponds to the recognition of an object) and ways of rendering (refers to virtual objects overlapping images captured). Because of this, it is indispensable the use of cameras that can capture the sequences of images of the world in real time, the recognition of any standard (tracking of markers) and the positioning (latitude and longitude) of some specific object, thus it is possible to occur the processing and plotting of virtual objects assigned to the real world (rendering).

On the developing of an AR system, there are several libraries [51] and APIs (Application Programming Interface) that assist and enable the development. Among these libraries, the most noteworthy [52] currently are OpenGL[9] (Open Graphics Library), Vuforia[10], Wikitude[11], VisionLib[12], OpenXR[13], ARCore[14], ARkit[15] and so on. Some of these technologies allow you to recognize the target object for 3D plotting by determining the exact location of the image, position and spatial orientation relative to the camera coordinate system and virtual system.

[9] `www.opengl.org/`
[10] `www.vuforia.com/`
[11] `www.wikitude.com/`
[12] `www.visionlib.com/`
[13] `https://www.khronos.org/openxr`
[14] `https://developers.google.com/ar/`
[15] `https://developer.apple.com/arkit/`

Although libraries facilitate development, AR systems require rendering accuracy for the virtual objects in the real plane, ie it is necessary that the virtual and real objects are aligned in a resulting world and to the user's perception [51]. Thus, there are some deadlocks that can impact the experience with AR such as, detection of markers for plotting objects, cameras with low resolution (quality) or even interference and reflection of light in the environment.

Render libraries have the function of converting the virtual world shape from the computer's internal database to the real world user experiences. Thus, these libraries must contain resources to render basic elements of a scene and other resources to enrich the display. For example, in a typical graphics library, in addition to having the ability to process basic shapes and specify vertices and polygon colors, the programmer can use the option of adding virtual lighting elements and overlay polygon photographs (textures) to make it appear more realistic. In addition, these libraries can support top-level graphics that give the option of functions such as descriptions of hierarchical objects, collision detection between objects [21], differentiation in tracking detection of specific markers, and the ability to perform various tasks in real-time.

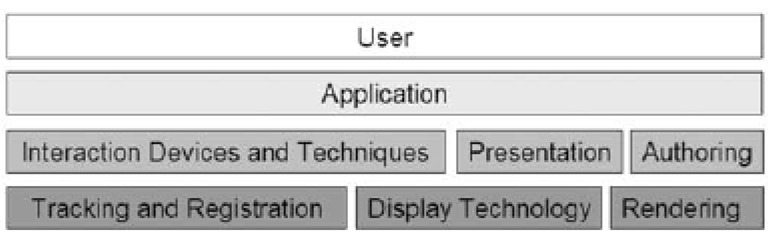


Figure 3.15: AR in blocks [48]

Some of the basic components of an AR system can be represented in block form (Figure 3.15). As can be seen, tracking and registration, as well as display and rendering technologies are fundamental components of the architecture. Above these are defined the most advanced modules, as the interaction devices and techniques, presentation and authoring. Only in the third layer where it exists the application and the interface to the user. Therefore, the most important requirements for creating an AR system are [47] [48]:

- **Virtual objects**: 3D object models created in CAD systems (Section 4.1.5) to increase the real-world view of the user (user perception);
- **Correspondence**: indicates the calibration accuracy between the real-world and the generated virtual-world;
- **Tracking and registration**: follow the user's pose to determine the point of view of the marker or relative object within the actual workspace. The registration is the sensation of getting virtual objects in the same space as the real in a stable way;
- **Real-time rendering**: a real-time display of computer-generated models matching real-world images of the environment. Plays an important role for these applications through faster and realistic rendering;
- **Processing**: this corresponds to the response time to camera movements and the accuracy of the recording between the image and the graphics generated (critical to the effectiveness of the system).

Orientation

The orientation of the devices can be considered as a technique for the development in AR since this has an impact on the generation of images [47] [53]. Through the combination of the magnetic field sensor[16] and acceleration sensor generates the orientation of the device.

The combination of the data from these sensors allows obtaining the pitch, roll and heading axes of the device (Figure 3.16). Thus, global positioning estimated and global direction [54] influence on quality's registration, as well as image overlays.

- X-axis (heading): measures the direction of the device is facing, where 0º or 360º indicates north direction, 90º east, 180º south, and 270º west;
- Y-axis (pitch): measures the slope of the device where the reading will be 0º if the device is flat, -90º if the top is pointed at the ceiling and 90º if it is upside down;
- Z-axis (roll): measures the lateral inclination of the device where 0º is flat on the back, -90º is facing left and 90º is the screen facing to the right.

[16] magnetic field compasses tend to be mismatched near metal objects (computer)

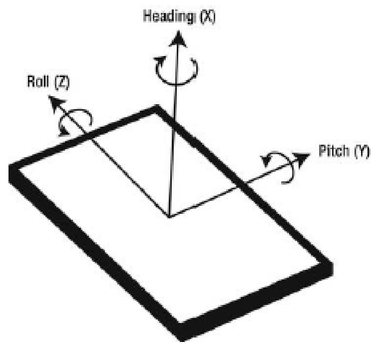


Figure 3.16: Orientation on a mobile device (handheld) [55]

Accelerometer

The acceleration of the device is given through the accelerometer along three directional axes: left-right (lateral X-axis), forward and backward (longitudinal Y-axis) and upward (vertical Z-axis). Figure 3.17 shows the axes of the accelerometer.

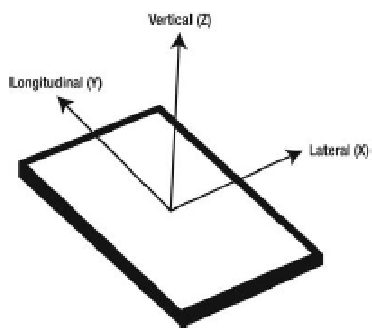


Figure 3.17: Accelerometer on a mobile device[55]

- X-axis (lateral): measures the lateral acceleration. That is, from left to right. Its reading is positive if you are moving to the right side and negative if you are moving to the left side;
- Y-axis (longitudinal): in the same way as the X axis, but it measures the longitudinal acceleration. Gets positive reading when the device is held in the same X-axis configuration, and a negative when it is registered in the opposite position;
- Z-axis (vertical): Measures upward acceleration and downward movement. Positive readings are upward movements and negative readings are downward movements.

3.6.3 AR System Types

Although there are several ways to make applications in AR, the most common is by using [51] markers that are similar to a QR code. These markers serve as a parameter for plotting 3D processed objects and also determine how far or how the 3D object is viewed. Thus, it is through the calibration of the camera [47] that calculation takes place to determine the position and perspective of the virtual objects. In this section, we will cover some of the types of augmented reality present, some concepts, and advances.

AR through Display (Handheld AR)

Augmented reality in monitors is one of the most common and simpler types for exhibition [50]. The experience is not said to be immersive and the augmentation itself only happens on screens and monitors, such as televisions or smartphones (also called Hand Held Displays [48]). Thus, monitors serve as windows for visualizing virtual world data, processing, memory, and interaction technology are contained in a single device [48]. Despite the simplicity, there is no additional high cost for its use, since the smartphone devices are consolidated in the market [2] and are in common use by the users. The Pokémon GO is an example of an AR application for smartphones (Figure 3.18).

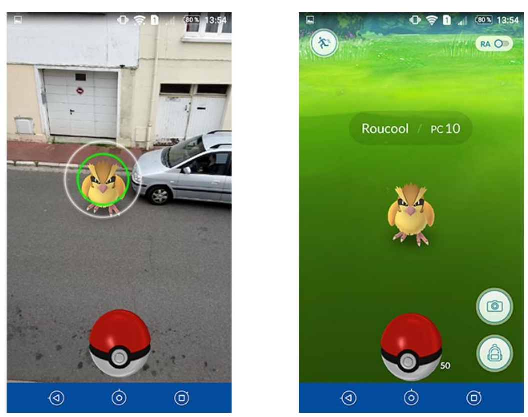


Figure 3.18: AR in mobile device [56]

AR through Visor (AR HMD)

Augmented reality through visor is used to give user experiences in a "straightforward" way. With it, the user can have a sense of immersion into the generated world, giving the possibility to have more realistic experiences from their own perspective.

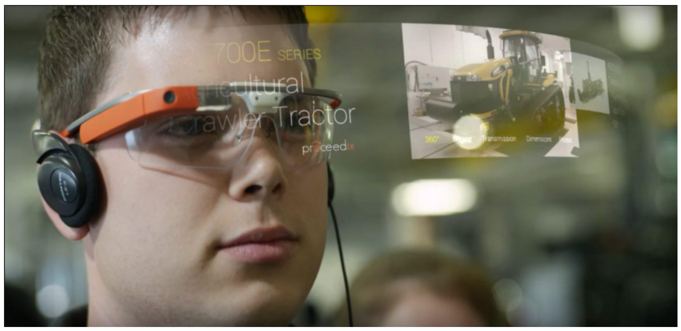


Figure 3.19: AR in Google Glass [57]

The main difference in this type of AR is that the augmentation happens and highlights differences depending on the current location of the observer (Figure 3.19). Although HMD are often associated with VR, they are also widely used in AR (Figure 3.20). In this context, the user can see the real world through a camera system coupled on the device that aims to reproduce the effective point of view of the user's eyes, that is, the viewfinder does not obscure the total view of the real world[17]

Spatial AR

Also called Projective AR [48], the Spatial Augmented Reality (SAR) approaches the same concepts of AR, but makes use of the projection of images directly to the real objects, that is, through spatial and mapping recognition of the ambient it creates the augmentation without having to rely on markers. It is said that SAR has a better understanding of the environment than traditional AR since through the use of other technologies, it also allows new technologies and approaches to be conceptualized for the evolution of this field (MR at the present moment)(Section 3.7).

[17] www.youtube.com/watch?v=JhCOqAMuegk

Figure 3.20: SAR with HoloLens device [58]

First introduced by Raskar [59]. SAR can "naturally" provides experiences with multiple users (similar to CAVE, Section 3.5.1). Most applications of this technique relate to a cultural context in which images are projected onto some surface, such as a desk, or a facade of a building. This new paradigm of the exhibition explores the virtual alignment of the virtual with the real giving possibility to use transparent screens and generate holograms [48].

3.7 Mixed and Extended reality

While VR uses the entire overlap of images, through the visible perception of the world, or the substitution of the real world for a simulated one [15] [16]. And AR allows projecting the virtual world into the user's world in real time, creating alignment between virtual objects and physical objects [59]. The Mixed Reality (MR) provides the interpolation of the worlds [60], between real and virtual (Figure 3.21), but in a way beyond the concept of traditional AR.

It can be said that AR is a subfield of MR, one of the divergences that differ one field from another is the terms of processing and technology. MR enables a broader

[18]the mixer metaphor: as used by the DJs nowadays, the two turntables are connected to a mixer that makes mixing of songs, in the case of MR, the mixture between worlds. By means of the one-dimensional X fader, it lets to choose several points along the X-axis between the Reality and Virtuality extremes.

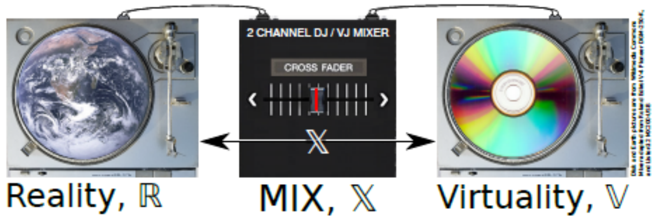


Figure 3.21: Mixer Metaphor[18][60]

environment recognition than the AR uses (world mapping). To do so, it makes use of more coupled cameras, in addition to infrared and other technologies to make the environment recognition and consequently the mixing of real and virtual. While it is possible to create applications in AR with just a simple cell phone or notebook camera, MR requires other sensors such as depth, side cameras to increase spatial sense and if used with devices attached to the head (Figures 3.22), specific technologies for lens.

(a) Microsoft HoloLens

(b) Magic Leap One

Figure 3.22: MR Devices [61] [62]

This not only provides a greater sense of immersion but also the interaction with the holograms becomes more natural. The visual processing of hand gestures, the recognition of voice commands, as well as the visual indication and spatial sound that some devices provide (Figure 3.22a) deal with a different interaction than usual AR allows. Some researchers mention that MR is the junction of VR with AR, although both concepts are different. In this way, other definitions are formulated to better address this vast domain and extent of these technologies.

3.7.1 Extended Reality

Regarding Extended Reality (XR), although this concept is not recent [60], can now be considered as the combination of all the fields: VR, AR, and MR. With XR, it is possible to "migrate between worlds" by recognizing the environment, using GPS, Artificial Intelligence, and joining with the Internet of Things (connecting various devices). This new approach allows researchers to present a new perspective on how the future might be[19].

Virtual worlds enabled by technologies, robotics, and other devices offer a variety of artificial experiences that are often physically dangerous or simply impossible in the real world [63]. Telepresence technology is an example, it allows humans to establish interactive relationships with their world in ways that transcend their scale, their spatial location and, often, their native biological capacities [60] [63]. By experimentally expanding the ways in which people can perceive, operate, and expand the limits and abstractions of their thoughts, they can extend what is virtually possible.

Although it may seem like tales of fiction, for the first time. Practical projects [60] with XR already exist, the assignment of new "senses" to humans is one of the examples. Thanks to the combination of other fields of study such as physics, biology, and chemistry, they allow the exploration of microwave waves visualization (Figure 3.23) and also could have touching sensation through haptic devices (Section 3.4).

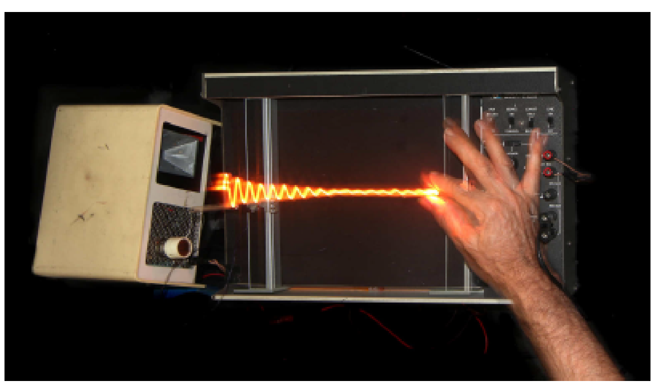


Figure 3.23: XR for wave visualization

In addition to these researches, other areas are benefiting from such technologies, the

[19]Hyper-Reality - https://www.youtube.com/watch?v=YJg02ivYzSs

mobility sector[20] is another example. With XR, it is possible to use reality enhancement to assist drivers in traffic, such as signage, streets, and avenues through maps and GPS, and assist in vehicles parking (Figure 3.24).

Figure 3.24: XR for vehicles parking

Despite the evolutionary phase that these devices carry is still in the beginning. Factors such as lenses for display, control of illumination and tracking movements, battery and connectivity have an impact on the progress of this technology and, consequently, on the daily participation of society. Therefore, the factors that still have an impact include [64]:

- **Display** - lens technology, more realistic visual content (rich in detail) and processing. Facility of change between total and partial virtual worlds;
- **Common illumination** - virtual objects in augmented worlds indistinguishable from real objects under the same view;
- **Motion tracking** - tracking for intuitive user head, hands and eye interactions;
- **Battery power and cooling** - battery with daily duration, recharging and compatibility with thin and passively cooled devices (without ventilation);
- **Connectivity** - ubiquitous wireless connectivity.

One of the factors that also deserves attention is the exploration of environment recognition [65] that can be optimized via software and/or hardware [66]. While the MR's devices (3.22) and concepts that these companies bring already let you recognizer the world

[20] `https://youtu.be/3uuQSSn07IE`

and make the fusion between, other concepts are being born to present better ideas. Another example that XR concept brings is the new way to do the recognition through a dynamical recognition of the environment in real-time [65]. As it allows to be faster than MR (first occurs scanning and then the recognition for the 3D objects appears) and also don't need a static or tidy ambient to do the mixture. It can plots the objects while recognizing the environment [65].

Thus, new projects that give initiative can end [66], because they aren't the best form of application or have the best cost of operation of this technology. However, XR can still be the future of the human-computer interface, since there is wide use of this tool and the impact it can cause, and also still evolving, areas of education, training [67], business transition [68] and other examples.

In reference to the target audience, such companies have different points of view. Microsoft announced that the next HoloLens 2 will be destined to the industry (Smart factory) [68] [69], besides their new device has standard apps [70] to the common user, this can be a way for the company doesn't lose some consumers, beyond the profits that the industries are investing. While Magic Leap is investing to the general consumer for using MR at home [71] [72] [73]. Therefore, the number of possibilities that this field can act and evolve deserves due attention, not only by the developers but also by Industry *4.0*. In the future, there may still be the possibility of downloading "themes of realities" in a virtual store, just as it already exists on applications for mobile cell phones and MR's devices today, this area can be the next generation of computing.

Chapter 4

Technologies, Materials, and Methods

This chapter aims to present the technologies, materials, and methods used to develop the application of this project.

4.1 Technologies

In this section, it is presented which technologies were used as well as the reason for their use.

4.1.1 Unity3D (version 2017.4.13)

Unity3D is a platform for game development, in other words, it is a Game Engine [74]. The Framework[1] encompasses a number of tools and libraries that support the creation of games, animations, and desktop/mobile applications (Android and iOS). Besides being a graphics rendering engine, it also has a physics engine that includes the detection of collision between objects and gravity simulation, it is also possible with the platform to

[1] framework: in software development, it is an abstraction that joins common codes among various software projects providing generic functionality. Unlike libraries, it is the framework that dictates the application's flow of control

create animations, assign sound effects and particles creation, editing and lighting.

It has more than one million users [75] and extensive documentation/online support for learning the tool.

The gaming engine consists of an integrated system for script-based development in C# and also brings the editors MonoDevelop and Visual Studio (in a Windows environment).

Some of the key concepts of the Unity3D platform for multimedia application development are:

- Hierarchy: consists of a panel that lists all objects in the scene in a textual way. Thus, cameras, lights, models of 3D objects and prefabs also belong to the panel. All of these elements are represented so that they can be selected and their properties can be viewed in the Inspector;

- Inspector: shows the details of an object [76] when selected in the panel. Lets you assign other object properties and settings;

- Scene: presents in visual form the images of the objects in the project described in the hierarchical panel. This panel allows you to move and change the level of detail of displayed objects as well as lighting properties, textures, wireframes or even combinations of properties of different objects;

- Project: all the assets used in the scene are grouped in the project folder. They are organized according to their type for example animations, audios, fonts of text, prefabs, other scenes, scripts, materials and etc.;

- Game Objects: consists of the representation of an object in a programmable scene, that is, assets, cameras, lights and other models that make up the project scene;

- Prefabs: are "prefabricated" objects [77], they function as object templates composed of one or more assets. It can be said that they are classes of software objects to be instantiated at run time in the form of being used in several contexts. For example, an animation prefab of a scene object that will be used in other ways, it is allowed to re-use internally of the code that constitutes this prefab and consequently to optimize the program.

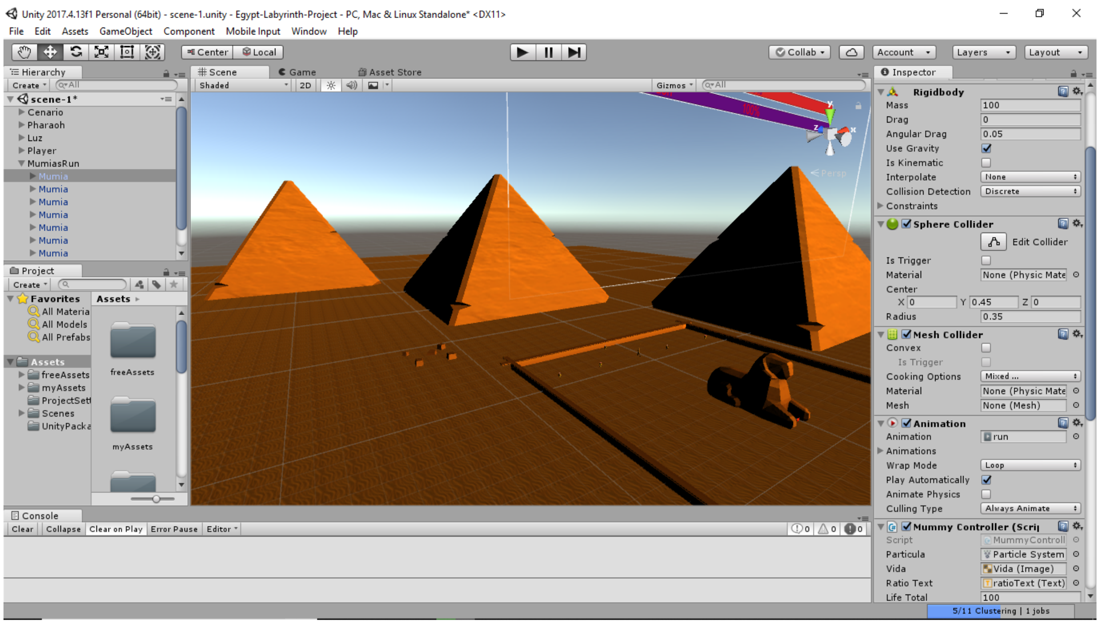


Figure 4.1: Unity3D Scene example[2]

4.1.2 C# Language

C# (C-Sharp) is an object-oriented programming language developed by Microsoft. It has similar syntax to other C, C++, Java, and Object Pascal languages, besides having an extensive online documentation[3].

The C# has begun to be developed in the late 1990s. The main factor to develop this language was to solve the impasses that the developers of the time faced. When a programmer needed to migrate to another language, it was necessary to learn not an only language like its libraries and new concepts.

As such, .NET (Microsoft Development Environment) is designed to work with multiple programming and sharing languages within the same set of libraries. The solution

[2]https://github.com/lucascviveiros/Egypt-Labyrinth-Unity3d-
[3]https://docs.microsoft.com/en-us/dotnet/csharp/

to this was the same as previously adopted by the Java language: a virtual machine for compiling binary codes.

Some of the functions of the virtual machine include memory management, threads, execution stacks, and so on. Because of this, the application is independent of the operating system, since its scripts are compiled primarily for the CIL (the intermediate language of .NET) and later interpreted by the virtual machine of C#: Common Language Runtime (CLR) [78]. In addition to the CLR runtime environment for all languages of the .NET platform, the C# virtual machine also isolates the application with the operating system, which makes it easier to isolate the compilation from interfering with the execution of other applications' operations system.

C# is also considered as a strongly typed and explicit declarative programming language, which becomes very useful for use in the development with linkage to the hierarchy of objects and manipulations of animations in Unity3D scenes.

4.1.3 Visual Studio 2017

Microsoft Visual Studio is an integrated development environment (IDE) for software development (Figure 4.2), the .NET Framework include also languages like Visual Basic (VB), C, CPP, F# (F Sharp) and in particular, C# (C Sharp) [79]. It can also be used for web development using ASP.NET, for creating websites, web applications, services, and mobile applications.

The process of running an application in Visual Studio occurs through the steps of passing application code (program source code) to the specific compiler (in the case C#, CIL) that will generate the intermediate code and later interpretation by virtual machine CLR. The intermediate code is then placed in an executable file (.exe extension) in the project folder. And so, the resulting compilation file generates the Assembly. After compiling, Visual Studio runs this assembly file on the C# (CLR) virtual machine and so runs on the operating system [78].

Visual Studio is a robust IDE for C# development and makes it a lot easier to demonstrate some of the library options that Unity3D uses like UnityEngine, System.Collections, UI, and even the Vuforia SDK 4.1.4 and MRTK.

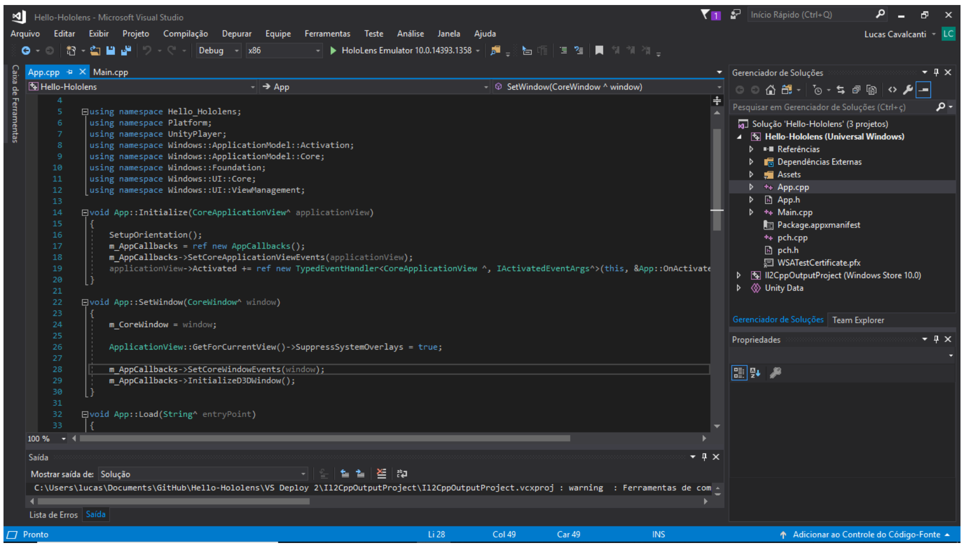


Figure 4.2: Visual Studio 2017

4.1.4 Vuforia SDK

Vuforia[4] is an SDK for Augmented Reality applications. Created by the American company Qualcomm (and acquired by PTC[5])), the SDK enables the development of applications in AR for mobile devices and desktops

The SDK utilizes computer vision techniques that allows to do the individual recognition and tracking of objects or patterns captured by the camera in real time. The SDK also allows, when linked to game engines like Unity3D, to position and orient virtual objects in real three-dimensional space, ie the viewer's perspective on the object in the virtual world corresponds to their perspective in the real world.

The main point for using the Vuforia SDK is that it allows the tracking and recognition of pre-configured markers to make the 3D model positioning in real time. In addition to

[4]`https://developer.vuforia.com`
[5]`http://fortune.com/2015/10/12/ptc-buys-vuforia-from-qualcomm/`

supporting Android[6], IOS[7] and Unity 3D[8] platforms, it also allows the development of a cross-platform application (compilation for different operating systems).

Figure 4.3: Vuforia SDK

Android SDK

To compile Unity3D applications for Android mobile devices along with Vuforia it is necessary to use the Android SDK 7.1.1 (Nougat) and API 25.

The Android SDK includes a number of utilities needed to help during development such as device role control, service integration, and the desktop emulator for application testing. In addition, it has online reading documentation that assists development.

Java Development Kit (JDK)

In the case that the application is intended for an Android operating system, it is necessary to also use Java SE (or JDK).

JDK is a development kit[9] for Java developers that also includes JRE. The tool assists in the development, debugging and monitoring of Java applications intended for the Android platform.

ARMarker

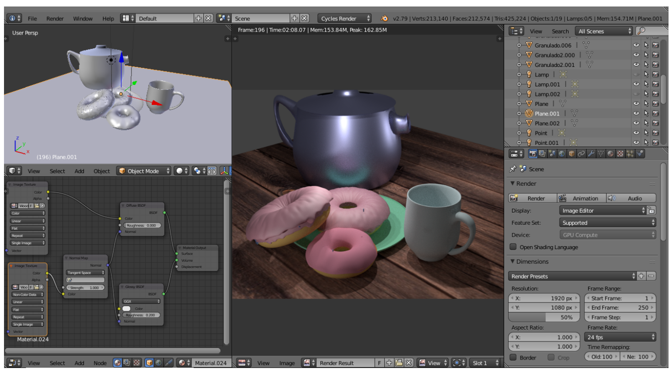
In order to have better accuracy in the recognition of the markers by the camera, an API can be used to generate Vuforia specific markers[10] (can also be used for ARToolkit).

[6] `https://library.vuforia.com/articles/Solution/Getting-Started-with-Vuforia-for-Android-Development`
[7] `https://library.vuforia.com/articles/Solution/Getting-Started-with-Vuforia-for-iOS-Development`
[8] `https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity`
[9] `http://www.oracle.com/technetwork/java/javase/downloads/`
[10] `https://github.com/shawnlehner/ARMaker`

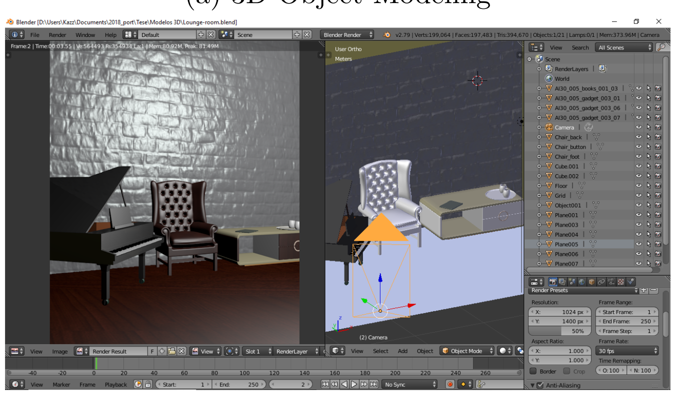
4.1.5 Blender 2.79

Blender[11] is a multi-platform suite for 3D creation (Windows, Linux, and Mac). The Open Source tool allows you to create scenes and interactive objects. It has options for 3D modeling (Figure 4.4), animation, video editing, and game development, besides allowing the import and export of other formats.

Blender is also developed in OpenGL, and currently uses Python [80].



(a) 3D Object Modeling



(b) 3D Environment Modeling

Figure 4.4: 3D Modeling in Blender Source: own authorship

One of the main reasons for using this suite is due to its formats being compatible with other CAD (Computer Aided Design) platforms that use extensions such as: obj, fbx, 3ds, stl, ply and so on. This also influences the import and export of 3D models built for the Unity3D platform, that is, tools matching.

[11] `https://www.blender.org/`

4.2 Hardware

The hardware used for the development of this project is an Acer personal computer, has 8 gigabytes of RAM, a 2.20GHz Intel i5 5200U processor, an NVIDIA GeForce 920M graphics card with 2 gigabytes of DDR3 memory and a hard disk containing 1 Tera Storage Bytes.

4.3 Mobile Device

The mobile device used for application testing was an ASUS mobile phone, model Zenfone3 X00DDB has 3 gigabytes of RAM, 1.4GHz processor, 32 gigabytes of storage, 16MP camera, 1920x1080 pixel resolution and Android OS version 8.1.

4.4 Microsoft HoloLens

The device version used was the first Microsoft HoloLens[12] commercial suite, has transparent holographic lenses, 2 16: 9 HD light engines, Automatic Pupil Distance Calibration, 2.3M Holographic Resolution, and 2.5k Holographic Density.

Also has 4 cameras of ambient comprehension, 1 camera of depth, a technology of Mixed Reality Capture (MRC), 4 microphones and 1 sensor of ambient light. The processor has 32-bit Intel architecture, a Holographic Processing Unit (HPU), 2-byte RAM memory and Windows 10 OS.

4.5 Methods

For the beginning of the development of this work, it is necessary to define how the training situation will occur, as the user will be instructed through an application in augmented reality to do the disassembly of the equipment. For this, it will be necessary to use the

[12] `https://www.microsoft.com/pt-pt/hololens/buy`

concept of "StoryBoards", which in principle, is directed to VR. This method constitutes the definition of the steps to be performed by the user with the AR application.

After defining which steps are required for each faucet model, it is then possible to create the 3D models in Blender. The 3D model generated depends on the defined parts that will be integrated into the whole object set. Once that the 3D modeling is finished it will need to export these models in ".fbx" format and then be imported into the Unity3D platform.

As defined in the StoryBoard, each part of the 3D object will represent the actual object, so both the animations and the Prefab structure itself can be defined in Unity3D (by choice the animations will be made on Unity3D instead of Blender).

At the end of the 3D modeling stage of the real objects, export, and import to Unity3D and generation of animations on the same platform. It can then configure AR settings with Unity3D using the Vuforia SDK.

This step consists first of generating markers using ARMarker, registering these markers on the Vuforia site, creating a key for development with the SDK, creating a database and checking the effectiveness of Vuforia recognition on these markers. Next, it can download the Prefab to later import into Unity3D.

After the development of these tasks, it is necessary to perform evaluations in the mobile application. The size of the marker (according to the tap model), the positioning of the marker on the tap and the effectiveness of the generated instruction should be checked.

The size and position of the marker have a great impact on the generation of augmented reality. As the 3D virtual object must overlap the actual object image seen on the user's mobile camera. For this, adjustments must be done to the size of the virtual object in the scene, as well as its positioning relative to the marker.

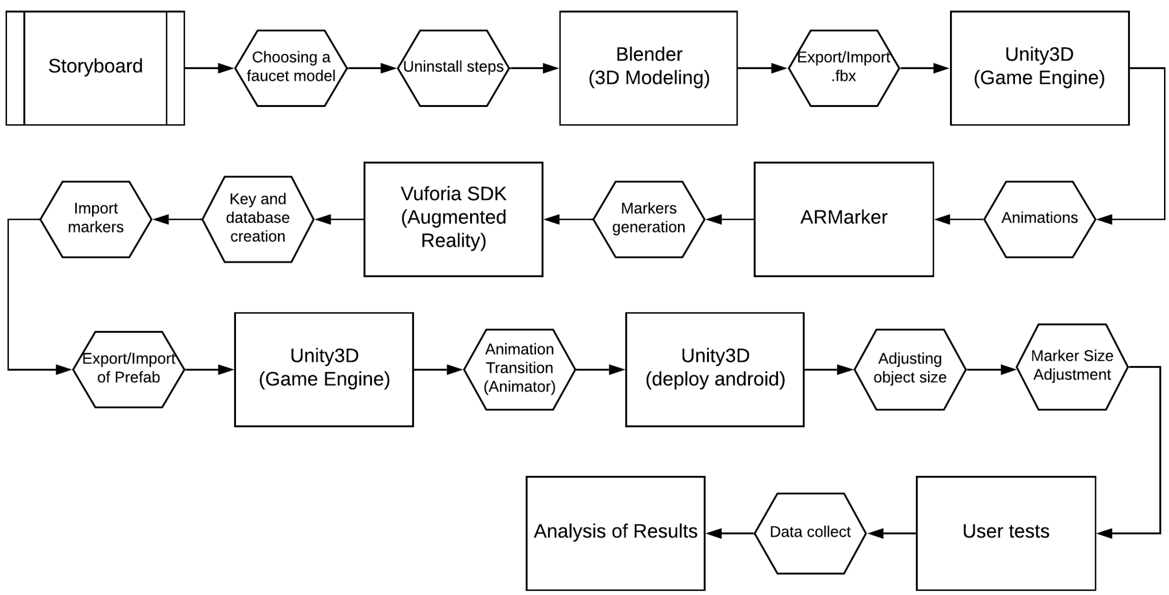


Figure 4.5: Construction Diagram

Chapter 5

Development and Implementation

This chapter will present the development of the application, as well as some of the impasses encountered during the elaboration process.

5.1 Storyboard

The Storyboard is one of the beginning strands for creating environments or scenes in VR. Allows you to use sketching in the form of drawings (in blank sheets or other applications [81] [82]) for creating immersion. Compared to "brainstorm", it is used before beginning the development and implementation of the project, so it is used in the creative process part, which becomes a progressive and abstract procedure, depending on the situation. Although it's more widely used for immersive VR applications, this concept can be used to create interactions with AR applications, since VR and AR scenes differ in real and virtual world participation.

In this way, the creation of Storyboard was chosen through drawings or hand-made sketches representing the faucet model, the parts that make it up and the following of steps (Figure 5.1) according to the assembling and disassembling process. So, as the technical knowledge about faucets maintenance operations is not from common sense, it was necessary that some instructions had to be learned before starting the Storyboard.

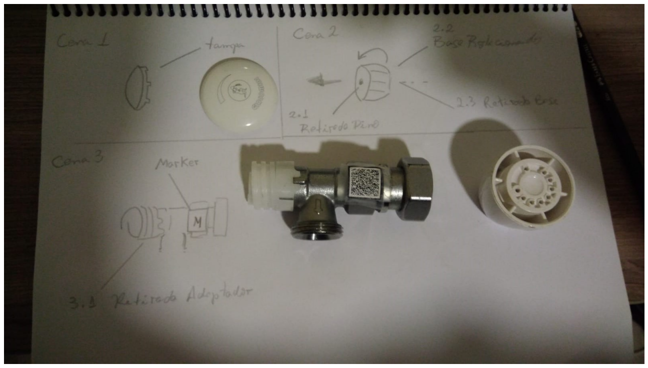


Figure 5.1: Storyboard Example

5.2 3D Modeling

To create the instructions for uninstalling faucets with augmented reality, it was necessary to create the actual physical samples in 3D virtual objects. Thus, the pieces that are part of the structure of the tap were all modeled in *Low Poly*[1]. The choice of this 3D modeling style was necessary because the level of detail is not as important to the user as this does not affect the effectiveness of past instruction and still saves on application processing by reducing the number of polygons or vertices.

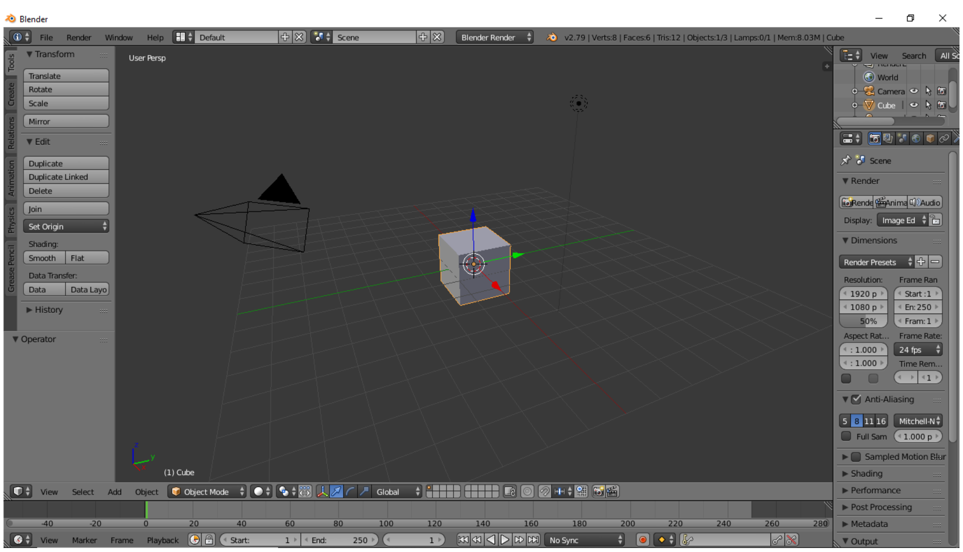


Figure 5.2: Screenshot Blender Init

[1]Low Poly - refers to modeling by similar assimilation of the real, that is, with few details or few vertices.

As shown in Figure 5.2 Blender's pattern objects are a cube, a camera focused on the object for rendering, and a light.

5.2.1 Blender Configuration

It is necessary to change the default rendering of *Blender Render* to *Cycles Render*, since GPU rendering becomes more efficient.

Other configurations before modeling can impact rendering time, so for this the number of *Tiles* was changed to the size of 512, as show in Figure 5.3. It has been noted that the number of *Tiles* has an impact on the rendering according to the number of objects to be rendered and according to the choice of CPU and GPU. It also can be said that with GPU rendering it is better for a larger number of objects and CPU for fewer objects, the adjustments vary depending on the video card and processor as well as the number of objects in the scene.

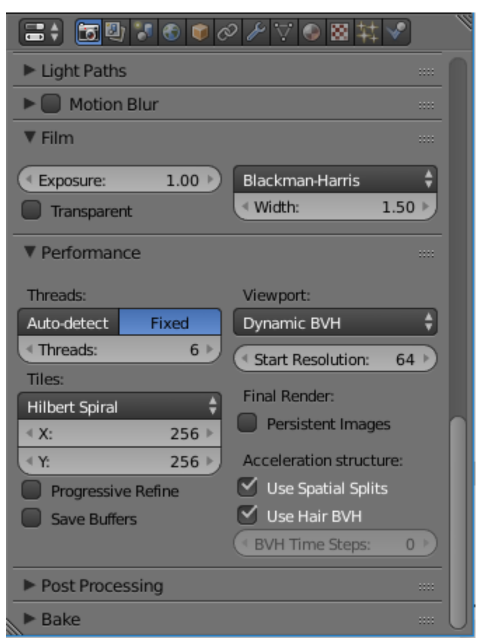
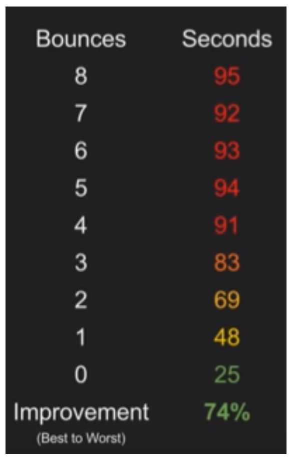


Figure 5.3: Blender Configuration

Light Bounces

Something that happens in the real world and also in 3D modeling systems, refers to light reflection. When the light hits a surface a reflection of the light for other surfaces happens. In 3D modeling tools, *Light Bounces* turn out to be a very costly process in terms of rendering time [83]. Thus, if some parameters are changed, a better rendering performance can occur, as can be seen in Figure 5.4.



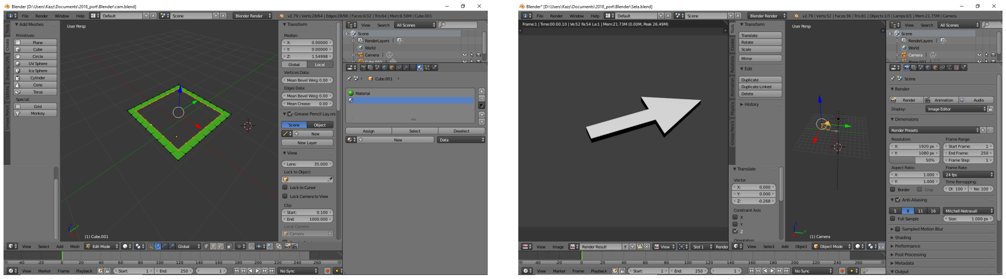
Bounces	Seconds
8	95
7	92
6	93
5	94
4	91
3	83
2	69
1	48
0	25
Improvement (Best to Worst)	74%

Figure 5.4: Light bounces [84]

If a maximum of 4 *Light Bounces* is defined there is no noticeable difference, after this, some objects can become opaque and end up impacting the final rendering of the object. Another point to note is that the *diffuse* and *glossy* parameter switched to 1 has reduced in less than 20 seconds in render time. As can be seen in Figure 5.4, there is a great improvement by changing parameters. Although you could make use of other configurations [84] in the Blender suite, these already allow you to start modeling objects without having as much impact on rendering time.

5.2.2 Models

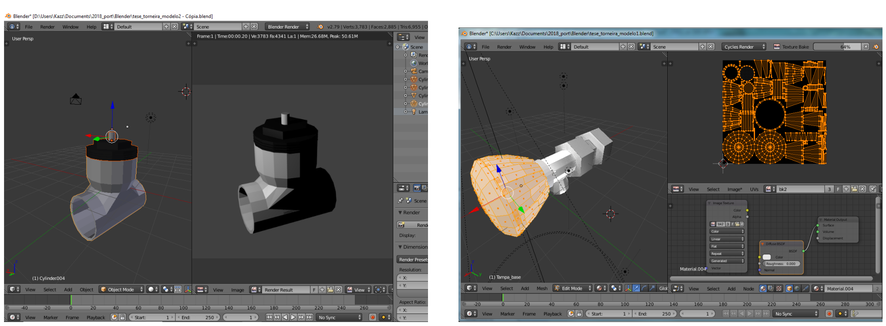
To begin the use of the tool, two simple models have been created, the first one indicates the recognition of markers as can be seen in Figure 5.5a and second one indicates the follow-up of instructions by the user (Figure 5.5b).



(a) Markers index (b) Instruction index

Figure 5.5: Instruction Models

As the project itself intends to make a case study of augmented reality applied in this sector, few pieces of the objects of physical faucets were modeled. For this, an easy-to-uninstall model was created according to Figure 5.6a and another model with a greater uninstallation abstraction as can be seen in Figure 5.6b.



(a) Faucet Model 1 (b) Faucet Model 2

Figure 5.6: 3D Models

In addition to the models mentioned above, it was possible to import an adapter model (Figure 5.7) through the extension **.sfx** created through the *AutoCAD* tool and export the same model to *3D Object*, or **.fbx**. This proves the compatibility and match between different extensions and computer aided design tools. It was also necessary to make a reduction of polygons, so that the model could impact on the execution latency of the application. It can be said that latency is also influenced by the mobile device system, as well as the CPU and GPU.

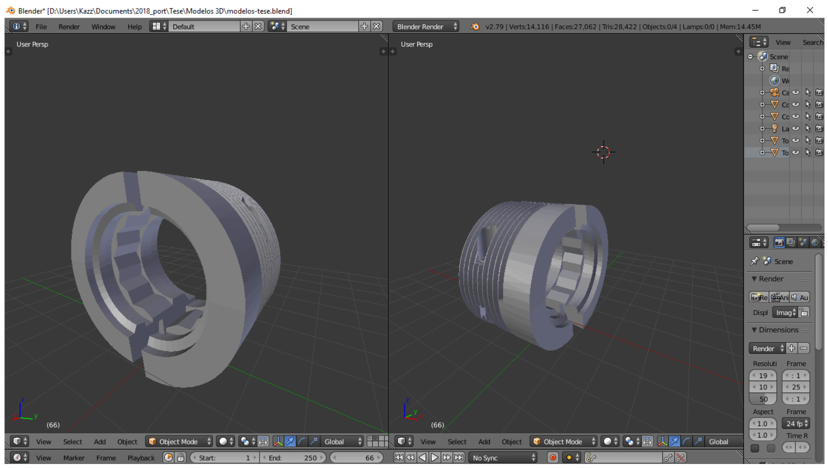


Figure 5.7: Imported Model Screenshot

5.3 Unity3D

After creating the models in Blender, it can start importing the models into Unity3D. This section presents the 3D Objects and Animation basics in the Unity3D platform.

5.3.1 3D Objects

After the creation of 3D objects in the Blender suite, these are exported and then imported through Unity3D in the *.fbx* format, as mentioned previously, in the Unity3D gaming engine (Figure 5.8). It was chosen to create the color and material in Unity3D and the ambient scene lights so that it was possible to make changes directly to the model, and made it easier for checking the Augmented Reality directly on Unity3D.

3D Animation Models

Before defining the user-manipulation instructions, it was necessary to define how the animations would be made. Animations can be done in the Blender suite along with object modeling, on Unity3D through the use of Animator along with Animation, or even through scripts (code-level[2]).

[2] Behind the Unity3D gaming platform are the libraries DirectX and OpenGL, so it is also possible to make animations in low-level codification, example `https://github.com/lucascviveiros/SpiderOpenGL`

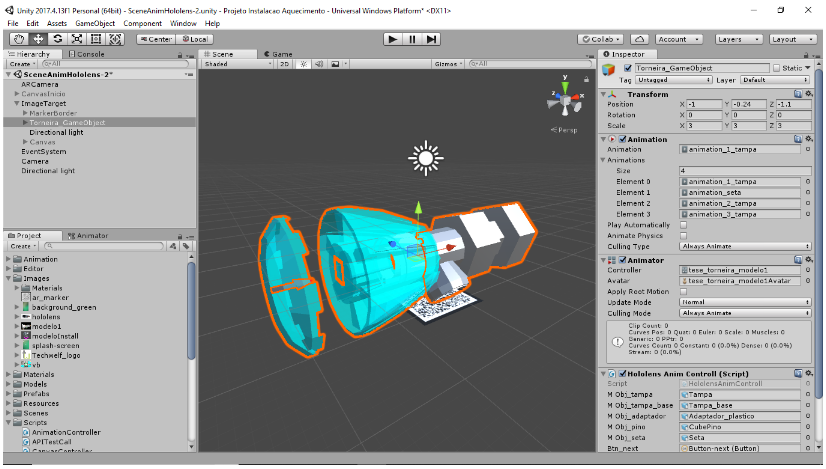
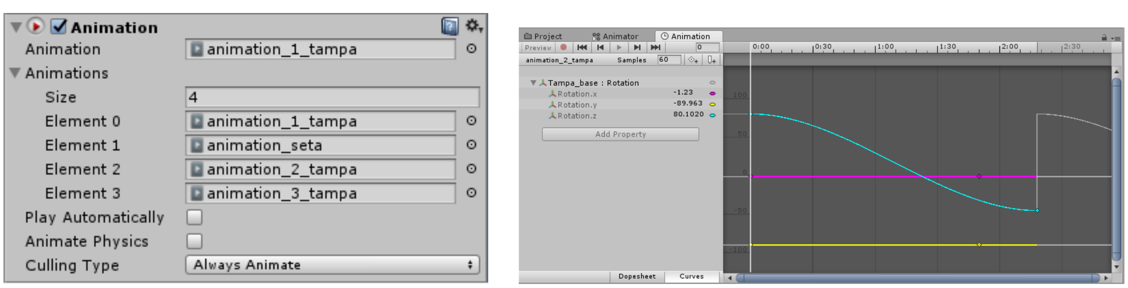


Figure 5.8: 3D Imported Object on Unity3D

It was chosen to use the Animator (and Animation) of the Unity3D platform, with management of the steps through C# code instruction. For that, was also used tracking of the animations through command entry that the user provides the application. So by giving the option to rewind the animation, to a statement or to move to the next steps.

To create Animations with Unity3D is necessary to use the animation controller (Animator). Animations by the platform can happen either through the definition of motion recording (Section 3.2.2) or by entering paremeters and changing the space curve along the time axis (Figure 5.9).

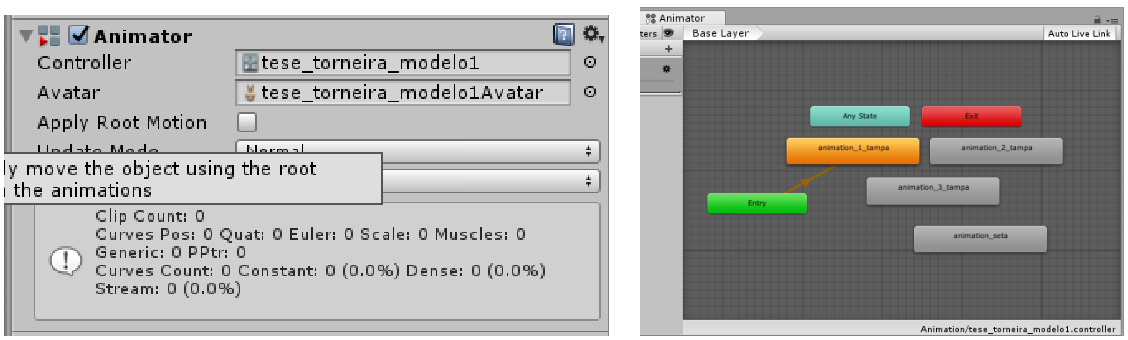


(a) Animation Inspector Unity3D (b) Animation Creation

Figure 5.9: Animations in Unity3D

As can be seen in Figure 5.9a, the animation elements are added in a list of animations, which don't need to be organized in chronological order since the control's being done via script. In Figure 5.9b, the variation along the x, y, and z-axes can be defined over time.

After saving all animations you can use Animator (Figure 5.10) to do the management. If animations have not been previously defined with Animator, you can not use them during program execution. Another point to be highlighted is that each model has its own controller. Therefore, other 3D objects (such as the instruction indicator) must have their own Controller.



(a) Animator Controller

(b) Animator Layer Unity3D

Figure 5.10: Animations Control

5.4 Vuforia SDK

This section explains some of the basics before starting with the Vuforia SDK. Therefore, this section covers part of the SDK architecture such as the camera, conversion of camera frames, what is a tracker, video rendering and dataset. And other points like the operation of the Trackable class, type of trackable and status, target definition, Virtual Buttons, and ARCamera.

5.4.1 Camera

The Camera access through the SDK ensures that each view frame is captured and passed to the tracker. Thus, during development, there's no need to use low-level code, it is only necessary to indicate when the catch (by the camera) should start and stop. In addition, the camera frame's automatically delivered in format and image size depending on the device used.

5.4.2 Image Converter

The image converter has the function of converting the pixel of the camera format to an appropriate format of OpenGL rendering, also serves to do the tracking. The conversion also includes a down-sampling which is a form of use for different camera resolutions.

5.4.3 Tracker

For tracking, Vuforia uses some computer-vision algorithms that detect and track real-world objects from the video camera frames (Device Tracker). Thus, according to the image that is captured in real-time of the camera, the algorithms can detect the targets. The results are then stored in a state object that is used by the video renderer and can also be accessed from the application code. The tracker can load multiple sets of data, but only one can be active at a time (Figure 5.11).

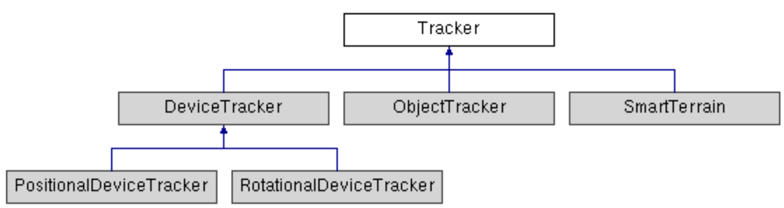


Figure 5.11: Tracker

The PositionalDeviceTracker establishes tracking of the device against the environment using visual detail of the environment in the camera's view to determine the pose of six degrees of freedom of the image (Section 3.2.2). For this, Vuforia Fusion[3] is used. This SDK provides a low-level code access (such as ARkit and ARCore), so it uses different technologies to access calibration, orientation, acceleration, and other device sensors (Section 3.6.2). The RotationalDeviceTracker is used for VR scenes, it allows to track the user's head when used of an HMD (inertial sensors)(Section 3.5.2).

[3] `https://library.vuforia.com/content/vuforia-library/en/articles/Training/vuforia-fusion-article.html`

5.4.4 Video Background Render

The rendering singleton of the *video background* renders the camera image stored in a status object. As a result, the performance of video background rendering is optimized for certain devices. It also sets the clip mode, so it uses the video background border to scale the objects.

5.4.5 Dataset

Vuforia uses dataset[4] as a set of trackers (collection of trackables) that the SDK provides when it is defined in its online management platform. Allows the app to load, enable, and disable dataset groups at run time. The enabled data set is then used by ImageTracker to locate and track the targets through the camera.

5.4.6 Trackable Class

The trackable class is used to identify, detect, and trace all objects in the real world in three-dimensional space, also lets you define some basic properties of a given object.

Trackables

The trackable[5] represents real-world objects followed by six degrees of freedom. Since each object is detected, its tracking definition is defined by the set of parameters that comprise the *target definitions* as type, name, ID, and status. The trackable are also defined by:

- Object_Target: a trackable single object type (real-world object recognition);
- Image_Target: an image-type trackable (single image);
- Cylinder_Target: a trackable consisting of images applied on a cylindrical or conical surface (curved image);

[4] `https://library.vuforia.com/content/vuforia-library/en/reference/unity/classVuforia_1_1DataSet.html`

[5] `https://library.vuforia.com/articles/Solution/How-To-Use-the-Trackable-Base-Class`

- Model_Target: a real-world specific object trackable based on the 3D format of the object model (CAD model);
- Multi_Target: a tracker composed of several images combined in a given spatial configuration (a box for example);
- VuMark_Target: succinctly, a trackable marker with Vuforia standard shapes.

Trackable names

Trackables must have names to be identified in the database. The identification is given by a string with characters defined by: letters from a-z (lowercase or uppercase). numbers 0-9 and [- _.].

Trackable status

The trackable state defines the state of the object. Since each object is updated as each pixel of the camera is processed.

- UNDEFINED - the tracked status is not set;
- NOT_FOUND - the tracker was not found, for example, the object that you want to track was not referenced or is not part of the database;
- DETECTED - object detected;
- TRACKED (monitored) - the object has been traced;
- EXTENDED_TRACKED (extended tracking) - the object was tracked in this frame using the extended tracking feature (set in Image Target Behavior).

5.4.7 Target Definition

In this sub section it will be explained how the type of target was defined. So first it was explored other forms besides the usual (marker in single image format) to do tracking. Therefore, firstly the object recognition option (Object Target) was explored and then the definition for using ARMarker.

Object Recognition

As mentioned earlier, the Vuforia SDK has the option of recognizing objects[6] from the 3D scanner software or from the models already created by CAD platform (Model_Target). When scanning the models with Vuforia it is also necessary to download and print the image as shown in Figure 5.12.

(a) Faucet Recognition (b) Metric Tape Recognition

Figure 5.12: Object Recognition Process

Therefore, it was used the scanner software in an attempt to recognize a faucet model (Figure 5.12a) to be used as the target of instructions. But, it was not possible to achieve the desired success, as only 35 points were recognized. This is due either to the object having a type of material with metallic reflection contrary to that specified by the documentation (the object must be opaque and with few moving parts). However, as defined early, this work had the intention to explore the best way to solve the case.

Other examples of objects that are difficult to recognize are a backpack, cap, and cellphone case. By making a comparison, the same scanner software was used to detect a metric tape object with plastic material and it was possible to recognize 282 points (Figure 5.12b).

[6] `https://library.vuforia.com/content/vuforia-library/en/articles/Solution/How-To-Use-Object-Recognition-in-Unity`

Marker Creation/Recognition

For building applications in AR with Vuforia you can use *Image Targets* which are images that the SDK can detect and trace. These images do not need special black and white regions or codes to be recognized. Although, in the documentation there are several topics in the related subject[7].

Vuforia uses a set of algorithms to detect and trace the features present in an image. After recognizing, it makes a comparison with the features of an object known by the database. Once detected, Vuforia scans the image along the camera's field of view. The resources extracted from the images are stored in the database and used to make comparisons at run time, so until five targets can be detected and traced simultaneously.

(a) Marker Generation (b) Vuforia Features Extraction

Figure 5.13: Recognition

Target Manager and Configuration

To add the targets created by the *ARMarker* API it was necessary to use Target Manager which is an online Vuforia tool that allows you to manage databases and generate licenses. It was chosen the *Device Database*, which is a target local database for images and objects stored on the user's device[8].

The target size indicates the actual size of the image in units in the Unity3D scene. Thus, during development it needs to specify in the target creation (Figure 5.14) or you

[7] `https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability`

[8] Vuforia also allows other types of database such as User_Defined and Cloud_Recognition.

can use the *XML* configuration file that is generated. The size parameter is important because the information that is returned in the trace indicates the object's scale when rendered in AR. In addition, it is recommended to scale the image to 320 pixels or wider, so you can avoid the impact of anti-aliasing[9].

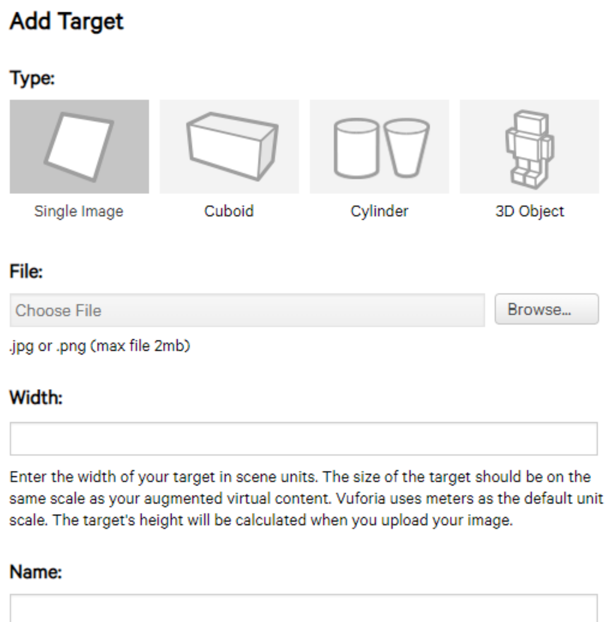


Figure 5.14: Add Target

5.4.8 Virtual Buttons

The virtual buttons[10] can be used as a way to have interactions with the application without having the need to use a touchscreen of the mobile device's for example. Its operation takes place through markers that are used as the basis for plotting virtual objects simulating a button in the physical environment. Thus, interaction occurs through real three-dimensional space, that is, the user can interact with a button created by the increase of reality by generating some action on it (in this case overlapping the marker)(Figure 5.15). So, the buttons were used to make the transition between the instructions steps.

[9] anti-aliasing : is a form defined in the images that result in optimization of detection and tracking of features.
[10] https://library.vuforia.com/articles/Solution/How-To-Implement-Virtual-Buttons

To create the buttons it was necessary to generate other markers through the AR-Marker API and later recognition by Vuforia, and import into Unity3D. The animation transition script was then set to use a virtual button event (IVirtualButtonEventHandler) and methods that trigger the button such as OnButtonPressed and OnButtonReleased. In addition, it was necessary to use a list to go through the event register (RegisterEventHandler) in order to be able to access each individual virtual button, ie identify which button event was associated and check the associated status. Unity3D also allows virtual buttons to be dynamically added and deleted at runtime.

Because just one script was created for the transition of the animations, it was realized that the script associated with the buttons should be in an another object (different from the objects related to the markers), as these objects when not “activated”or rendered at runtime do not trigger the script. Another way to solve, was to use the “extended” option of ARCamera. This configuration module enabled by the ARCamera allows even when the camera of the mobile device is not focusing on the marker, and consequently rendering of the virtual objects, can keep the script active. So by this configuration, the object remains present on the scene. That is, it manages to keep detection of the object even though it is not within the camera’s view, however other ways also could be done.

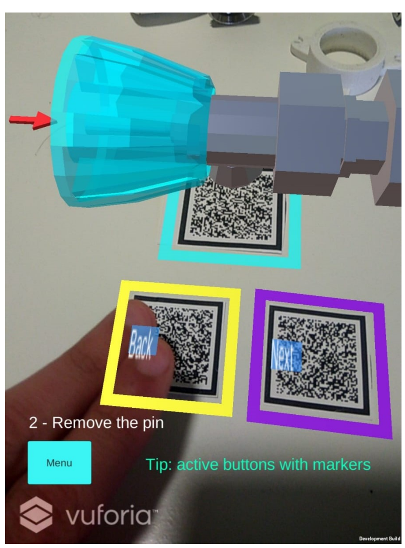


Figure 5.15: Virtual Buttons

5.4.9 ARCamera

Unlike the standard Unity3D's Camera, ARCamera is configured with a solid black background which allows rendering transparency during execution. Thus, for its use other settings on Vuforia SDK must be defined (Figure 5.16). In Vuforia Behavior script, the application license key (generated on the vuforia site) is used and other settings such as the set of amount of runtime trackers, and the database and camera's direction (front or back). Other point to be highlighted is the type of device to which the application will be compiled. Moreover, the type of *World Center Mode* that also implies in how the virtual world behaves in execution. There are 4 modes: FIRST_TARGET (the world is static and relative to the first target), SPECIFIC_TARGET (same as above but the target is explicit in the editor), CAMERA (the position and orientation vary depending on the actual target recognition distance) and DEVICE_TRACKING (position and device rotation are relative to the world) most used in VR scenes.

In the case of mobile devices is used the standard HandHeld, when used with HoloLens this parameter should be changed to Device Eyewear or HoloLens (depending on the version of Unity3D). When the Device Eyewear is used to HoloLens the camera becomes inverted at the run time with Unity3D.

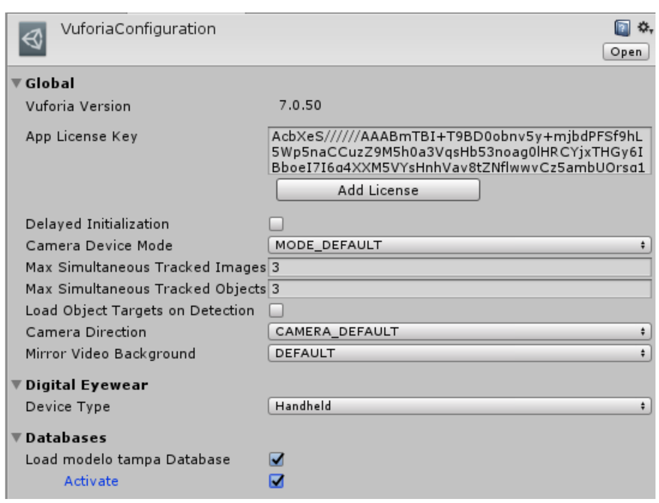


Figure 5.16: ARCamera Vuforia Behaviour

5.5 Solution Diagrams

According to the basic concepts presented previously and from Vuforia's documentation, this section presents how the application works through diagrams and sequence of events e main classes (5.5). More specific parts can be found on the documentation.

In the Diagram 5.5, it is highlighted that firstly a conversion takes place, from the camera frames to pixels and after to matrices. In ImageConverter, the SDK uses conversion to OpenGL ES, turning in low-level code. Then, the UserDefined indicates the conversion that occurs from the previously defined markers (ignoring what is not part of the camera's view) and thus, succinctly, the tracker occurs. In latter case, the detection and tracking of ImageTargets and VirtualButtons occurs, and finally the rendering of 3D objects, animations and AR through the camera.

In the Sequence Diagram 5.5 is possible to have other approaches. When the user chooses the scene the *Awake* method starts *ARController*. This uses a *Camera Listener* and links all Vuforia Behavior events and cycles with *ARCamera*, both to start and control other behaviors. So, through *DeviceController*, that is, the *Device Class* (also the class that change between worlds[85]) uses the *RenderingPrimitives*[11], and DeviceTracker [86] to give the *orientation* and *calibration* (Section 3.6.2) of the cellphone, as the *scale* and *spatial* relations (related to updating the conversion matrices [87]). Vuforia uses personalized calibration to optimize the accuracy of static registration content.

ARCamera also has a link with the behavior of the trackers (Trackable Behavior). That is the base class of the SDK that controls the *detection* and *tracking* of all types of trackables [88] (EnableTracking, TargetSearchResult, StartRecognition, IsRequesting, etc) (Section 5.4.6). It is also the class responsible for merging the targets by the ImageTargetBuilder Class (encapsulates all functionality needed to create a user defined target on the fly). Also cotrols the changes from the targets (ImageTarget and Virtual Buttons) through events (ITrackableEventHandler, IVirtualButtonEventHandler) [89] and related

[11]Provides a set of unified function calls that are designed to work across all AR/VR display devices, including hand-held, head-mounted (monocular/stereo) and wearable mobile devices.

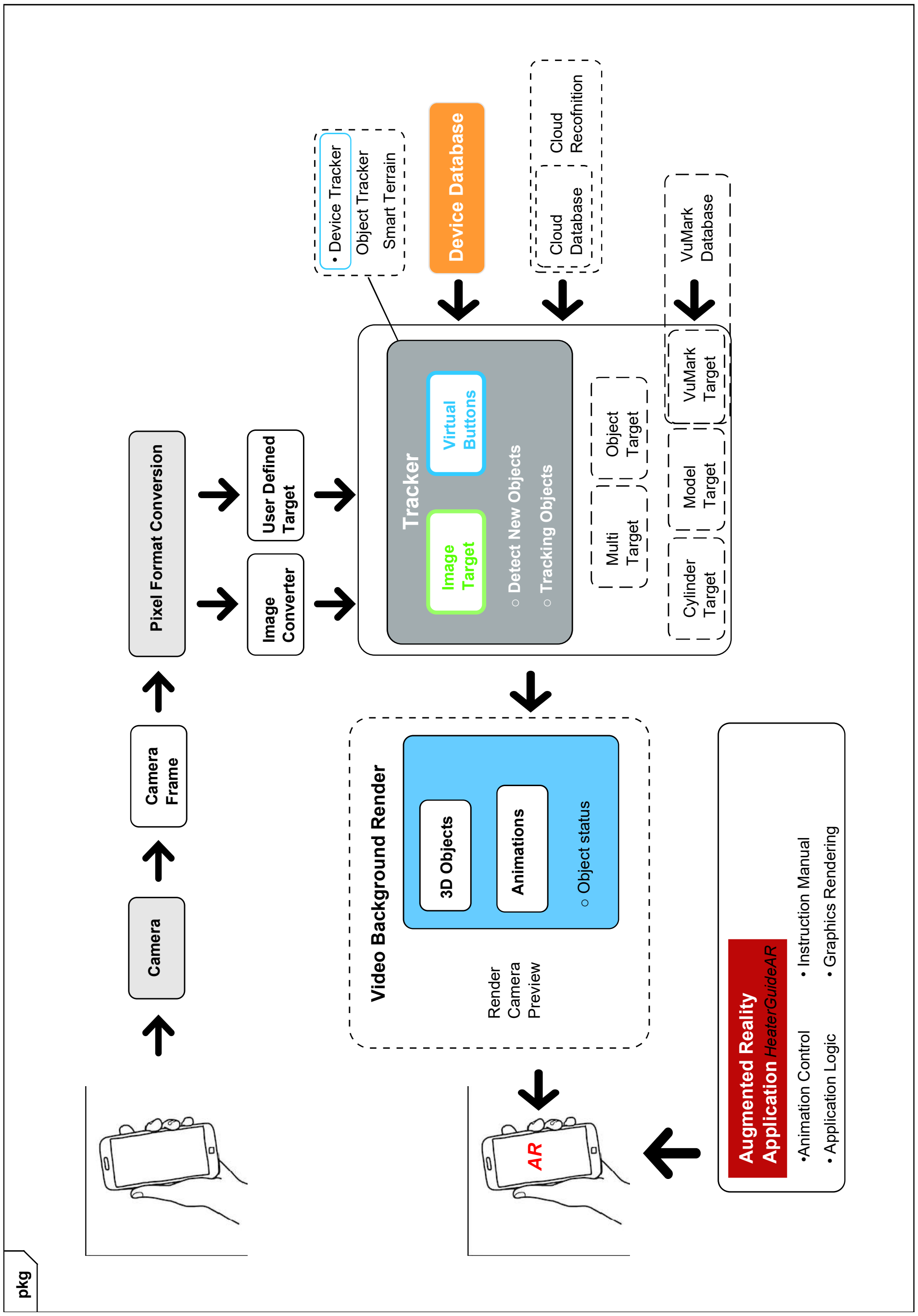
to the methods (OnButtonPressed/OnButtonReleased). For this project [90] it was created just the OnButtonReleased to control the Animations.

In view of the DatabaseLoadARController (local databases of images or objects stored on the device) controls the dataset [91] (ActivateDataSet, CreateDataSet, DeactivateDataSet, and etc.). And has a link associated with the VideoBackgroundRender that access and defines the clipping mode through the clip sizing of the 3D objects in scene (SetClippingMode), as the shadders[12] (SetMatteShader) [92].

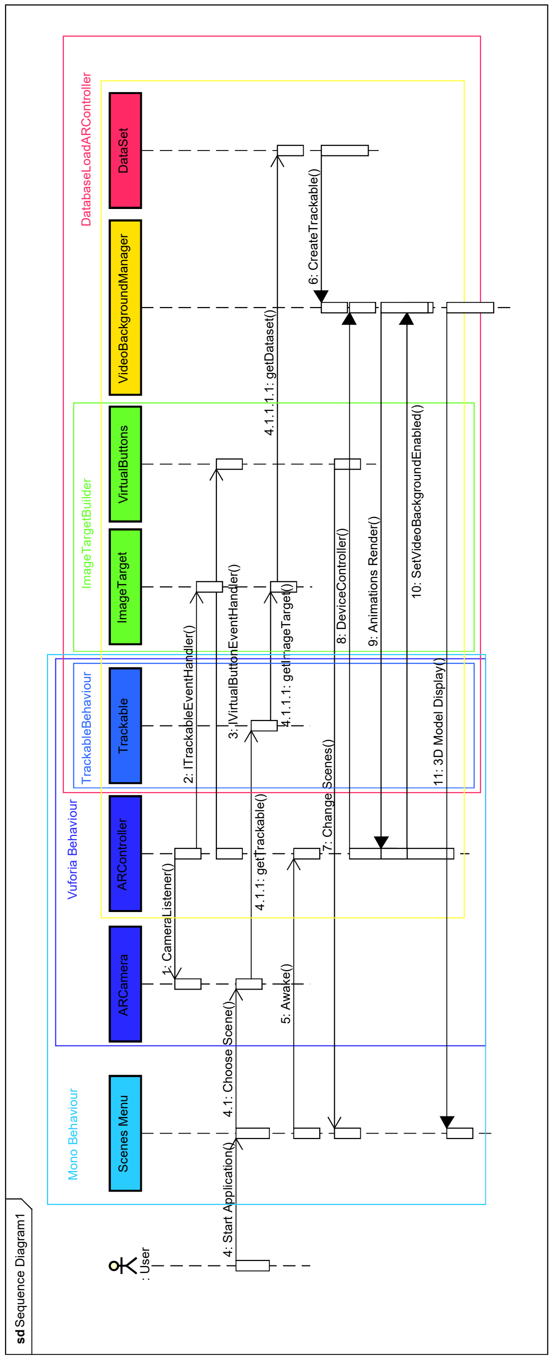
In ImageTargetBuilder is defined the Dataset, created at runtime by a user-defined destination (or searched at run time by the cloud) also responsible for generating the correspondence between worlds. Lastly, the creation of trackable (CreateTrackable) and activation and deactivation of the render to the device's screen (SetVideoBackgroundEnable).

[12] Shaders are commonly used to produce lighting and shading in 3D modeling, are also related to the GPUs and the realistic effects of the 3D object.

5.5.1 Application Diagram



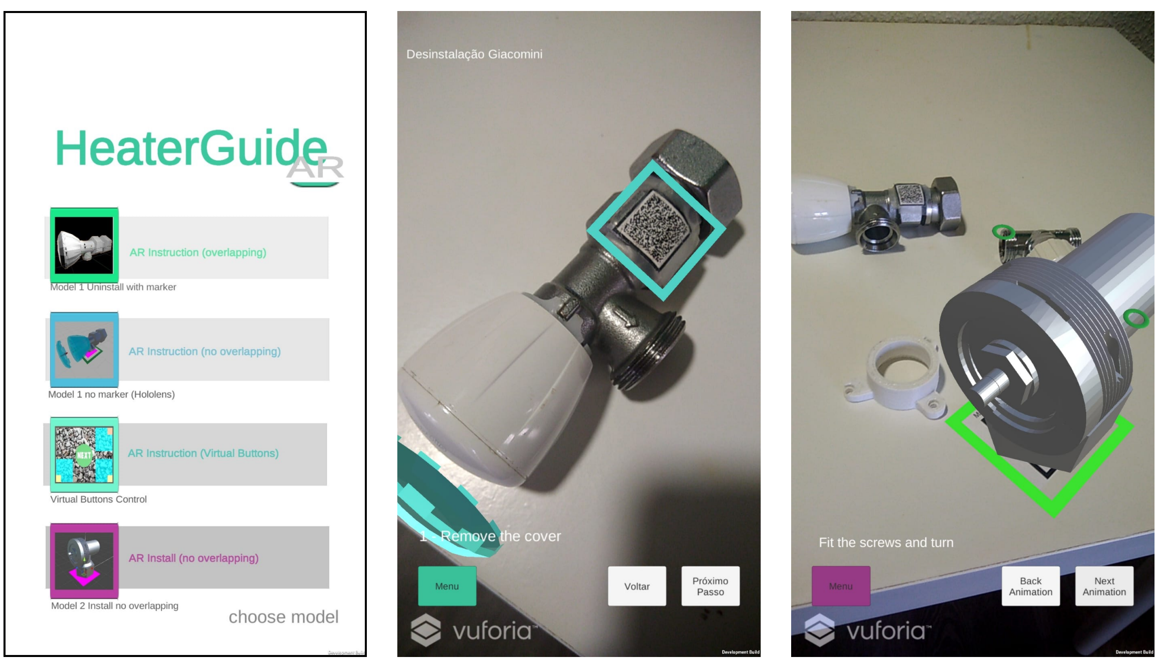
5.5.2 Sequence Diagram



5.5.3 Scenes Control and Application

In the scenes control, a menu has been created (Figure 5.17b), each scene is executed according to the button that is triggered. The first scene refers to the uninstall instruction of the faucet model of the Giacomini brand. The second scene refers to the instruction without overlapping of the real model. The third scene refers to the instruction with virtual buttons in AR (Section 5.4.8). Finally, the last scene refers to the instructions for assembling the model without overlapping the real one.

It's worth saying that the second scene was also created to HoloLens device, but as it was needed to change all the project and the canvas (Section 5.7), this menu is just for the mobile application.



(a) Scenes Menu (b) First Scene (c) Last Scene

Figure 5.17: Augmented Reality in Android

5.6 Comparison with Wikitude SDK

Besides the concepts and development with the Vuforia SDK, the Wikitude SDK was also used to make comparisons between the tools. As the concepts and mechanisms that both tools use are similar, only the comparison between performance and development of the solution will be presented.

Wikitude[13], as well as Vuforia, provides a number of tools that assist in programming with AR. It has an online database manager and recognizes and detects features of the target. It also has extensive compatibility with various technologies (Figure 5.18).

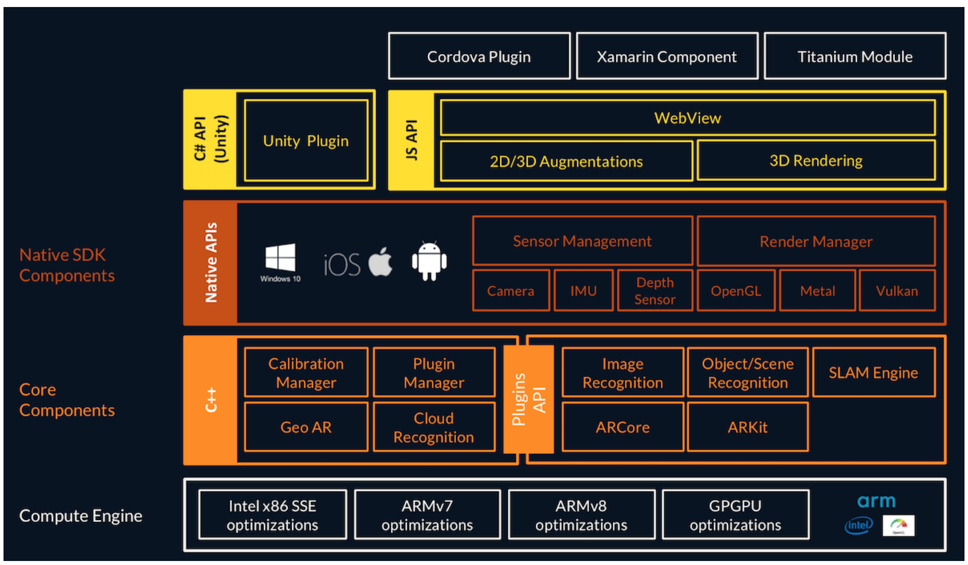


Figure 5.18: Wikitude Architecture

Regarding the development, the SDK is similar to Vuforia, it also requires a key created by the online platform, it gives the option to recognize various types of targets, such as objects, images and plans, and also allows cross-platform development (like Android and iOS). Moreover, is necessary to make external configurations in the Unity3D platform. Because of this, you should make the download/import of the SDK and recognize the trackers on the online website, and then check the file sent to your email.

It was noticed that during the development of the scene, it was possible to use the same marker created in the ARMarker, as well as the the Prefabs previously used with Vuforia

[13] https://www.wikitude.com/

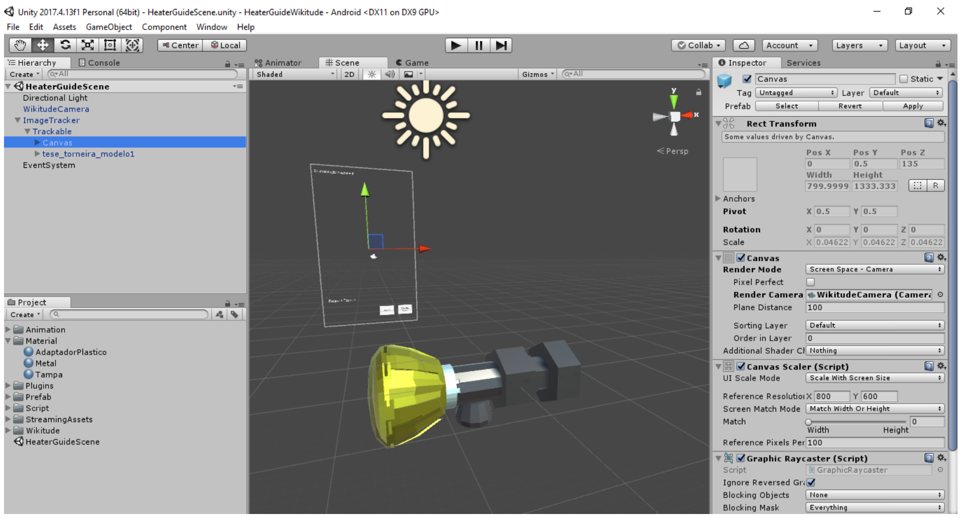
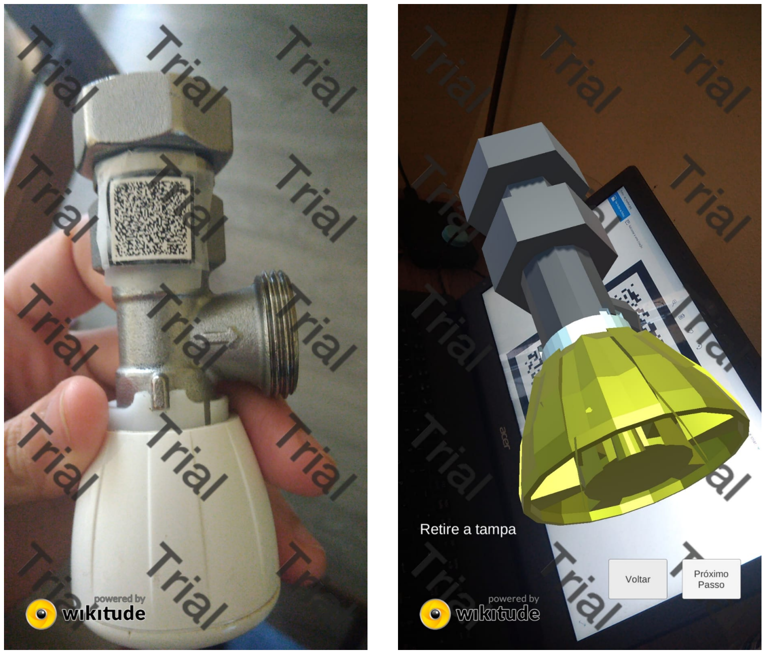


Figure 5.19: Unity3D Wikitude Scene

as: 3D models of the faucets, animations, the positions and scales for precise overlap, the UI interface of Canvas, control of animations and part of the scripts. Moreover, it was necessary to use WikitudeCamera (similar to ARCamera) and ImageTracker (similar to ImageTarget) available in the imported package. In addition to these, the package also provides the feature of ObjectTracker, InstantTracker and CloudTracker.

Regarding the methods, there are similarities between both technologies but performance was not the same with the application for Android. Could be noticed that the Wikitude SDK was not able to recognize small markers by the camera of the device (as this would be one of the requirements for augmented reality on the heater faucets). In spite of the wide compatibility that the SDK allows (Figure 5.18) its trial version is no better than the SDK that Vuforia provides. With attention to the non-recognition of small markers (Figure 5.20a) and the execution that contains texts in the rendering of the frame's camera (trial version) (Figure 5.20b). This can be the main reason for the Vuforia SDK can be installed together with the Unity3D tool (default configuration) and 65% of AR applications use Unity3D with Vuforia [93].

As a result, the recognition accuracy analysis of the markers are unequal between the tools (Vuforia recognizes markers faster regardless of size and distance). Other point is that the trial version appears to be fairly stable to the marker (similar to Vuforia) with



(a) Marker Unrecognized (b) Big Marker Recognized

Figure 5.20: Wikitude Tests

respect to the background video rendering with the 3D objects. However, one point that deserves to be highlighted is that when the tracker was lost and then recovered by the camera, the animation of the 3D object model did not render to the default position, that is, there was no correspondence and precision between worlds when the tracker was lost. This may be solved by changing or adjusting the SDK script, if allowed, or by purchasing other versions of the SDK such as Pro version for €2490 a year[14]. Even so, Vuforia standard license is free for the community of developers and yet with no watermarks.

5.7 Vuforia and Microsoft HoloLens

The purpose of this project was to create an augmented reality application for mobile devices. However, was also exploited other forms of compatibility between cross-platform devices. Thus, it was explored whether the Android project could through some other configurations have a link with the MR Microsoft device: HoloLens.

[14] https://www.wikitude.com/store/

During the investigation it was possible to verify that other requirements[15] should be satisfied such as: Windows 10 OS, Universal Windows Platform (UWP), HoloLens emulator (although the tests have been performed on the real device), and the Mixed Reality Tool Kit MRTK[16]. it has been noticed that by using the MRTK, the Microsoft device allows three-dimensional mapping of the environment. So there would be no need for using markers. Even so, it was possible to use part of the Unity3D project settings with the Vuforia SDK, and other definitions[17] (Figure 5.21).

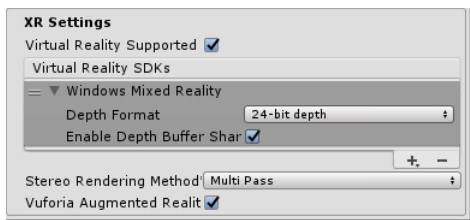


Figure 5.21: MR Configuration Unity3D

It was noticed that with MRTK and HoloLens it could be possible to exploit a lot of things that with AR mobile app could not as, make use of hand gestures recognition, voice commands through speech recognition (classes such as KeywordRecognizer and Dictionary), change of cursor or "gaze" and ambience recognition (or spatial perception). Recognition of hand gestures happens through image processing that provides the option of manipulating with holograms such as moving, dragging, rotating (Section 3.2.2) and action clicking on the object. Thus, by making use of all the scripts[18] created for the project with HoloLens, it was possible to do adjustments directly in the 3D Object, like using voice commands (next and back) to change the animation, increase and decrease the size of the 3D object (bigger and smaller), also stop and start the animation (freeze and play) as to rotate the model (right and left), all basic commands used were through voice recognition.

[15] https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools

[16] https://github.com/Microsoft/MixedRealityToolkit-Unity

[17] https://library.vuforia.com/content/vuforia-library/en/articles/Solution/Working-with-the-HoloLens-sample-in-Unity

[18] https://github.com/lucascviveiros/HeaterGuideHoloLens/tree/master/HeaterGuideHoloLens/Assets/Script

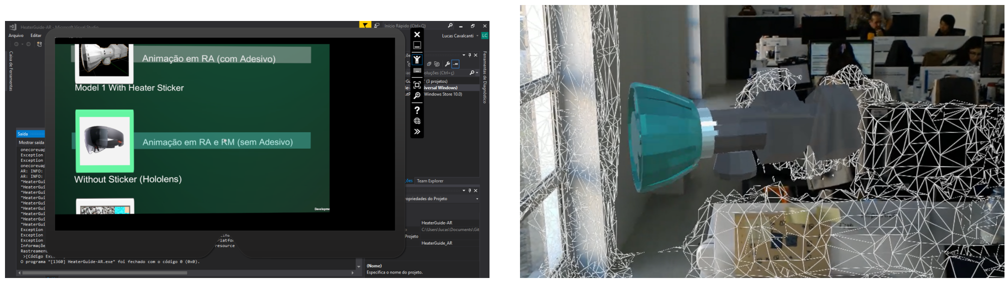
Figure 5.22: Student using HoloLens

Other point to be highlighted is that with the spatial perception and hand gesture it could change the place where the 3D object rest in the real environment.

Moreover, other configurations also had to be done such as the “UWP quality settings”, the Unity3D “Build Type”, the SDK definition for using HoloLens, and other details such as “Scripting Backend” and the “Publishing Settings” (to have access to the microphone, connection of internet, spatial perception and so on). It was noticed that just like the Android device deployment, in HoloLens it is necessary to enable the developer settings by pairing between devices in the “Update and Settings” option. Another point to be highlighted was that in the change of devices in Unity3D the Canvas object of the scene underwent a change (Figure 5.23a), so it was necessary to make adjustments for another Canvas template. Canvas is an area that encompasses the entire user interface such as text strings, buttons, and images. So, an adjustment had to be made in the menu scene as well as the size of the interfaces and strings.

5.7.1 Other details

Regarding to deployment details, both Android and HoloLens devices were successful deployed, yet some deadlocks were encountered. As Unity3D platform lets you make “Build and Run” right on your Android device, something that was not expected occurred



(a) Canvas in Emulator (b) Spatial Perception

Figure 5.23: HoloLens MR Application

in one of the cases, during the implantation and indication of success by the console of the Unity3D. It was verified that the application had not been implanted in the device, even though it had indication of success. So, for this case it was necessary to do another compilation check through the external Android Studio executable: adb.exe (Figure 5.24).

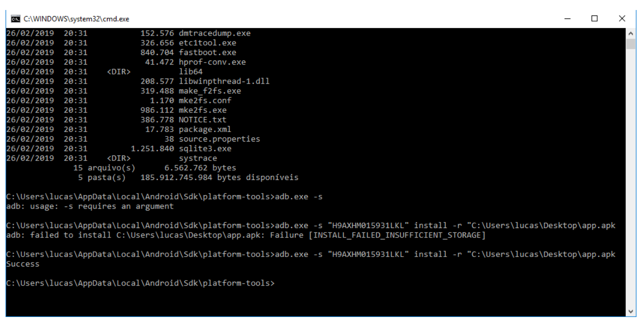


Figure 5.24: Android Deploy

To understand and fix the error it was necessary to view the details indicated by the Windows command prompt together with the Android native development application. After identifying that was a simple error caused by not having space on the device, it was possible to perform the compilation and deployment successfully.

In view of HoloLens with Vuforia, it was noticed that for configuration adjustments, from building the solution on Unity3D to opening the application in Visual Studio and compiling it to the device (or even in the emulator), there is a lot of time spending for the first time, even more for emulator. This can also be emphasized if you need to do adjusting and exporting/importing the 3D models again between Blender and Unity3D. Even so, with time you get used with all the tools.

Another point to be highlighted is that it was needed to do a project adjustment, in the part of the animations. It was necessary to update every animation in Animator through Inspector. So, when checking, it should be noted if the “Motion” option is set as “None”. If this occurs it will not be possible to perform the animations. To fix this you need to import all animations into Animator. The same thing may happen when you activate the “Legacy” option of the animation in the “Debug” tab. When the Legacy is changed, all animations in Animator must be imported again.

Chapter 6

Testing and Evaluation

The objective of this chapter is to demonstrate the tests performed in the application, as well as the results, the questionnaire carried out with the students and the problems encountered.

6.1 Application test

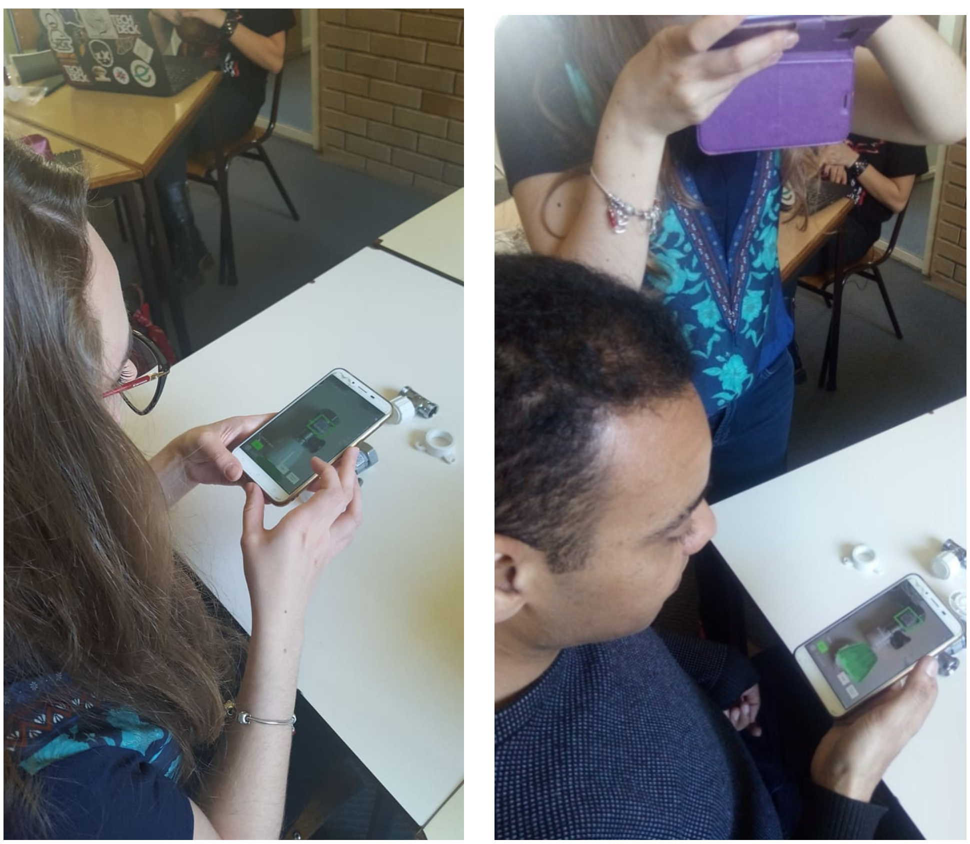
To verify the solution, a trial was performed with the mobile application. The evaluations were carried out by 16 students of several courses of the Polytechnic Institute of Bragança.

During the trial, it was possible to use different sizes of markers (and different sizes of 3D virtual objects), and the same standard marker chosen was used for augmented reality steps, which refer to the creation of virtual objects superimposed on the actual model of tap and the animations in AR without overlapping the actual original model. This was an alternative to the impasses encountered regarding the use and operation of instructions at the same time by some users. This is, to do operations like watch the animation, focus on the marker and do the instruction.

When the user understood the solution in augmented reality, they would have to make use of both hands to carry out the instruction because the available faucet model was not fixed in a place, which made it difficult to move during the operation.

It was noticed that during the instruction, the user took more time to actually perform the procedure on the model than to have the understanding by the application. This is due to technical difficulties that prove the difficulty of the subject in question. A guitar pick was also available that could assist in removing the cap from the faucet base in one of the uninstall steps.

Another point was the need to use a glue tape so that the marker appears to be an adhesive although it has been printed on sulfite sheet. In addition, another detail is that items such as the pick (or other aid objects) and the sticker could be provided by the company upon receipt of the new product. After the tests, it was possible to get some of the students to submit a questionnaire regarding the experience of the interactive manual.



(a) User 1 (b) User 2

Figure 6.1: Users testing application

6.1.1 Quiz

The GoogleForms tool was used to create the quiz employed in this investigation. It allows the creation of online questions and graphics to explore the results. The questions formulated aim to demonstrate if the application corresponds to the central idea of the beginning of this project. That is the augmented reality can really abstract the difficulty of instruction.

Although the application was deployed both on the mobile device and MR device HoloLens, it was only possible to test with the students through the mobile device. Because the Microsoft device belongs to CeDRI[1] (Research Centre in Digitalization and Intelligent Robotics) it could only make use of the device inside the laboratory and only to people who had permission.

Even so, some questions were created to test the use of the mobile application and to verify if the user already had some experience with another interactive manual and if was able to uninstall and install the faucet model. As well as whether would prefer to pay a technician to do the service or whether it was easy to use the application.

The first (Figure6.2) and second (Figure 6.3) questions were formulated to verify if the user who had never used an interactive manual through AR succeeded or failed in make use and maintenance of the faucet.

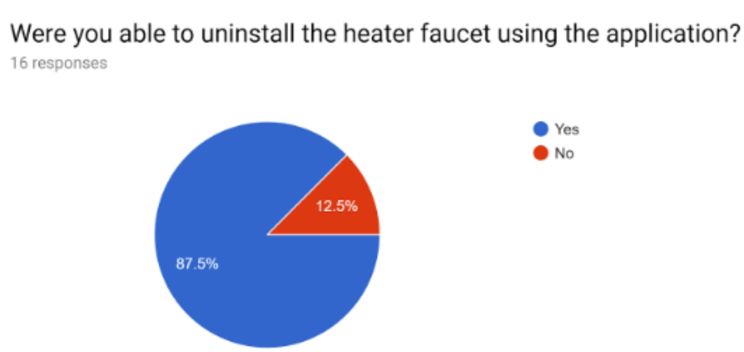


Figure 6.2: Uninstall Results. Source: Own authorship

[1] http://cedri.ipb.pt

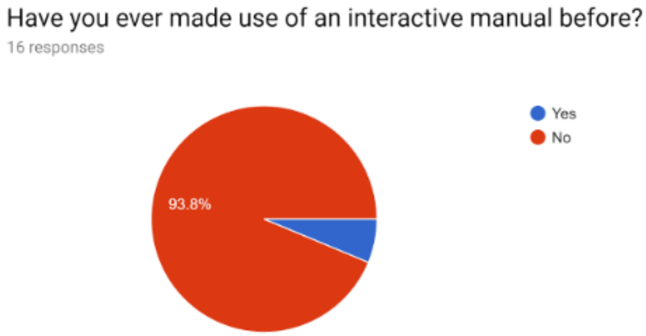


Figure 6.3: Verification Results. Source: Own authorship.

As can be seen in the chart of Figure 6.2, only 2 people (12.5%) were unable to uninstall. It was noted that people who were unable to do the uninstallation had difficulty removing the cover from the base of the tap, so they did not use any object of assistance (This emphasizes that instruments for operation can actually help in uninstallation).

Another factor worth mentioning is that even people who did not use or had any experience with AR apps before were able to make use of the application and maintenance without the need of a technician according to Figure 6.2. Even unsuccessful users recommend using Figure 6.4. This also can be emphasized by the chart of Figure 6.5.

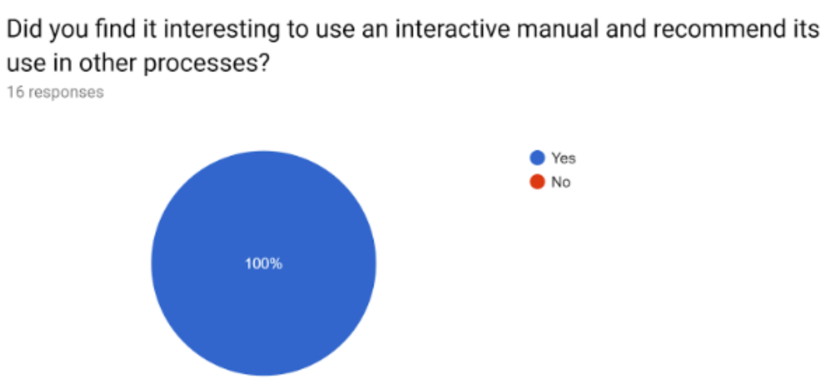


Figure 6.4: Interest/Recommendation Results. Source: Own authorship.

Although the test done with the users are simple, hiring a service to perform such an operation is not encouraged (Figure 6.5). This could be extended for further testing, validation with more people, more ages, more complicated models possible and other factors that can be taken into account. Thus, the determination of the results of contracting or not outsourced service would be better validated.

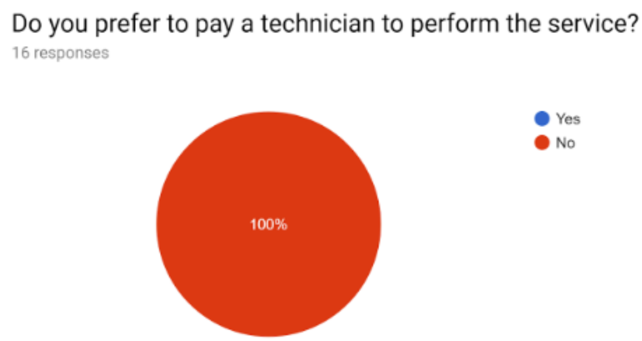


Figure 6.5: Preference Results for hiring outsourced service. Source: Own authorship.

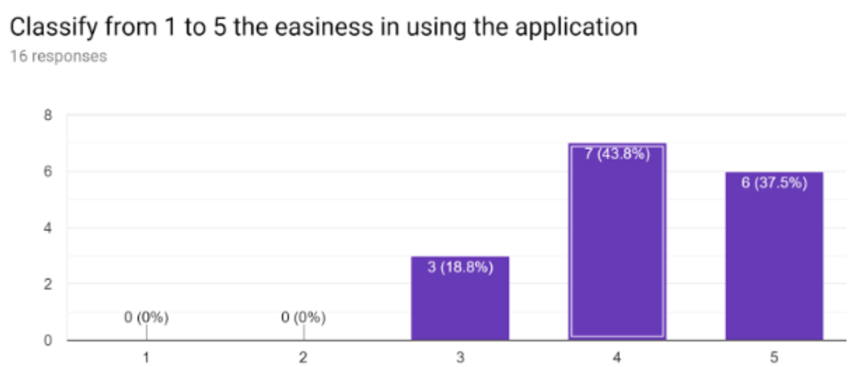


Figure 6.6: Easiness Classification Results. Source: Own authorship.

Another point that the questions verified was how ease was to use the application. According to the chart in Figure 6.6, most students scored the facility with the options 4 and 5. It is possible that students who chose 3 did not use the instrument provided for maintenance, yet it was a relatively low frequency compared to the total. It is worth noting also that since the students who have proposed to participate in the experiment are young and likely to use day-to-day technologies, they have experience for using mobile applications. Perhaps people who are older or who do not use mobile applications as the student would not have the same results. However, it is worth noting that the use of technology, especially mobile, is increasing, which makes the interactive manual initiative become more common over time.

Finally, the last question (Figure 6.7) addressed if users prefer using an AR app or a written instruction manual. More than 90% of the users prefer to make use of an AR application than having to read instructions (even though the research was done only with university students), highlighting the reason for the growth of this sector [2].

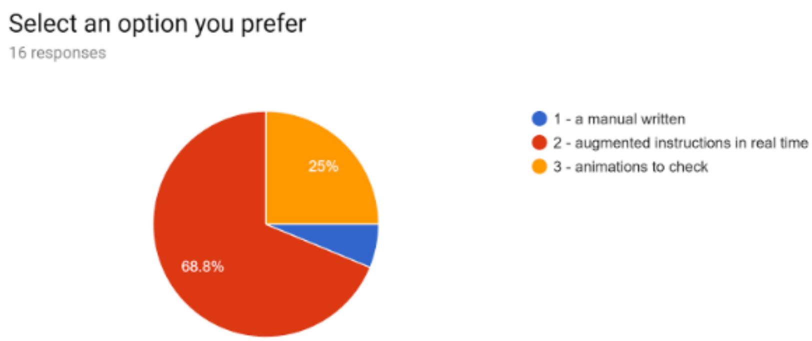


Figure 6.7: Preference results between written manual and AR instructions. Source: Own authorship

Despite the tests with the students were done only with a mobile device, from the experience with the Microsoft device, it was noticed that there is some difficulty in the beginning since the recognition and formulation of gestures with the hands is not something so usual yet. That is, there is no other source or form of use of this same concept (by another device). Still, over time it can become more common to use.

6.1.2 Problems and adjustments

Some of the problems encountered are associated with the operation part of the instructions, together with the attempt to create a maintenance kit by nonstandard tools, that is, by the operator handling it was noted that the adhesive did not remain in the determined position so that it was an a adaptive factor. When the user followed part of the instructions the adapted adhesive gave off and discouraged the continuation of the process, although the chosen model was one of the easiest one. Thus, several adjustments were necessary, mainly by the operator change. If the solution were actually adopted by a company in this industry, certainly the materials used, such as the adhesive, would have to be standardized and validated according to the type of material and location of the equipment to which it would be attributed.

Another point was that depending on the chosen environment the light was another factor of importance since this has influence in the recognition of the marker and creation in the reality augmentation. Perhaps the use of the camera's light from the cellphone would help in the process. The user's mobile handling was also another issue, as the camera's "Auto_Focus" mode was set during application development [94]. The option to change camera focus types at run time may be another way of solving the case of blurring of vision in-camera recognition with the marker, this is also influenced by the light and shadows of the environment.

Perhaps among the greatest impasses was the use of reality augmentation with the virtual button, this discouraged the use, also because it is not something usual by the users and the distance between the display of the camera and the mobile phone with the virtual three-dimensional object button. It may be that gestures with increasing reality become more common through the adoption of technologies such as HoloLens since the equipment itself greatly influences the exploration of these concepts and provides a much more immersive reality and experience gain compared to use with the mobile device.

Finally, the last detail was the marker size, since for the specific overlap of the actual faucet object chosen and the augmented reality itself, became necessary to create a very small marker in size (emphasizing that recognition only was made possible by the Vuforia SDK). The generation of markers and their recognition classification by Vuforia is another point to be highlighted, as well as the recognition of real three-dimensional object points by Vuforia's scan has a great impact on the generation and tracking of the increase of reality. If the marker used had not been rated with maximum recognition by the Vuforia platform, it was likely to impact its tracking by the application.

Chapter 7

Conclusion and Future work

Constantly, new concepts and technologies are created and united to solve several use cases. While areas related to VR, AR and MR are still expanding, it's up to us researchers and developers to track their growth and distinguish what each concept originates.

Understanding AR concepts is a key task as well as the current methods and materials related to the changing of each technology. When exploring the content of the subject the prototyping and creation itself can add more value to the final solution.

This work allowed investigating the principles of Augmented Reality and its aspects and comparing concepts and different technologies. The difference between VR, AR, MR, and XR also allowed creating a summary of development with Blender and Unity3D for applications in AR (mobile devices) and MR (Microsoft HoloLens). For those who have an interest in the area, they can make good use of the content in this work.

With respect to the development kits mentioned above, it was noted that through their use with Unity3D, they allow the development of the project to be directed to the final solution, in conformity of the adjustment between the real and virtual world. This also includes the compatibility among other tools used for modeling three-dimensional objects, generating markers and classification. Furthermore, also includes the experience with the augmentation through virtual buttons, gestures, voice commands, and the spatial mapping.

Regarding the solution for the maintenance instruction of the heating systems, this can be chosen with AR since through the results that this research demonstrated and with the current advances of this technology in the market can become more and more adjusted over time. That is, it can become a common sense by society. As a result of prolonging the research in this sector would extend to other taps models, other technologies and customer support kits in addition to other age groups. Moreover, investment in the solution is essential under the contract option of the outsourced service, its pros, and cons.

It is also worth mentioning that adjustment in the light reflection has effect by some objects and influence in the recognition of these, as well as the impact caused to the instruction solution in augmented reality. Another factor is that the use of markers adds another formalization to the customer, for the placing of the marker sticker on your heating system. Other point is that the influencing factor of adoption of this technology is caused by the usuality that certain applications provide, so games or forms of entertainment generate more affinity for society and encourage more its use.

Through the explorations that this work has allowed, such as definitions, junction of technologies and requirements, it is deduced that the possibilities that such technology provides are vast. In the future, the fusion between worlds, the way people interact and even the assignment of new senses (perception of reality) [60] may be increasingly associated with high-quality graphics and computer-processing. In consideration of the evolution, institution and the way in which the example of smartphone technologies influenced the way of relating and interacting.

"Holograms are actually objects of light that are attributed to our perception through technology, making possible the abstraction of thought forms into reality, it changes the way we see the world" [95].

Bibliography

[1] S. Mandal, “Brief Introduction of Virtual Reality & its Challenges”, en, vol. 4, no. 4, p. 6, 2013.

[2] C. Sanderink, *Virtual Reality Augmented Reality Hype Or Serious Business*. 2017.

[3] A. V. Dos Reis and B. S. Gonçalves, “Interfaces tangíveis e simuladores de veículos: Avaliação do honda riding trainer”, pt, *Design e Tecnologia*, vol. 4, no. 07, p. 20, Jul. 2014. [Online]. Available: `https://www.ufrgs.br/det/index.php/det/article/view/191` (visited on 09/20/2018).

[4] H. Chan, H.-W. Kim, and S. Goswami, “An Alternative Mechanism for the Cognitive Fit Theory in Spreadsheet Analysis”, en, p. 13,

[5] *Funcionamento das torneiras termostáticas*, `http://www.leroymerlin.pt/Site/Fazer-e-facil/Tutoriais/Funcionamento-das-torneiras-temostaticas.aspx`, Accessed: 2018-11-11.

[6] M. Żmigrodzka, “Development of virtual reality technology in the aspect of educational applications”, en, *Marketing of Scientific and Research Organizations*, 2017. [Online]. Available: `http://minib.pl/en/development-of-virtual-reality-technology-in-the-aspect-of-educational-applications/` (visited on 09/05/2018).

[7] J. W. V. De Faria, E. G. Figueiredo, and M. J. Teixeira, “Histórico da realidade virtual e seu uso em medicina”, pt, *Revista de Medicina*, vol. 93, no. 3, p. 106, Sep.

2014. [Online]. Available: http://www.revistas.usp.br/revistadc/article/view/103403 (visited on 09/04/2018).

[8] *Sensorama morton heilig*, http://www.mortonheilig.com/InventorVR.html, Accessed: 2018-10-05.

[9] T. Mazuryk and M. Gervautz, "History, Applications, Technology and Future", en, *VIRTUAL REALITY*, p. 72,

[10] C. Arth, R. Grasset, L. Gruber, T. Langlotz, A. Mulloni, and D. Wagner, "The History of Mobile Augmented Reality", en, *arXiv:1505.01319 [cs]*, 2015. [Online]. Available: http://arxiv.org/abs/1505.01319 (visited on 11/22/2018).

[11] H. R. Hildebrand and N. A. Menezes, "Cinema Interativo: A Imagem Viva", pt, p. 6,

[12] *Sound in virtual reality and video games*, en. New York, NY: Springer Berlin Heidelberg, 2017.

[13] J. Jerald, "The VR Book: Human-Centered Design for Virtual Reality", p. 79,

[14] N. Owano, "Magic Leap moves beyond older lines of VR", en, p. 2,

[15] M. K. Zuffo, "A convergência da realidade virtual e internet avançada em novos paradigmas de tv digital interativa", p. 90, 2001.

[16] M. Zyda and T. DeFanti, "Praise for Understanding Virtual Reality: Interface, Application, and Design", en, in *Understanding Virtual Reality*, Elsevier, 2003. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9781558603530500197.

[17] A. V. Netto and L. D. S. Machado, "Realidade Virtual - Definições, Dispositivos e Aplicações", pt, p. 33,

[18] J. Jankowski and M. Hachet, "A Survey of Interaction Techniques for Interactive 3d Environments", en, p. 30,

[19] J. Peterson, "Virtual Reality, Augmented Reality, and Mixed Reality Definitions", en, p. 4,

[20] E. Murphy-Chutorian and M. Trivedi, “Head Pose Estimation in Computer Vision: A Survey”, en, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, Apr. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/4497208/ (visited on 11/21/2018).

[21] A. B. Craig, W. R. Sherman, and J. D. Will, *Developing virtual reality applications: foundations of effective design*, en. Burlington, MA : Oxford: Morgan Kaufmann ; Elsevier Science [distributor], 2009.

[22] M. Turunen, “Error Handling in Speech User Interfaces in the Context of Virtual Worlds”, en, p. 8,

[23] G. P. Rodrigues and C. D. M. Porto, “Realidade Virtual: Conceitos, evolução, dispositivos e aplicações”, pt, *Interfaces Científicas - Educação*, vol. 1, no. 3, p. 97, Jun. 2013. [Online]. Available: https://periodicos.set.edu.br/index.php/educacao/article/view/909 (visited on 11/09/2018).

[24] A. Coppens, “Merging real and virtual worlds: An analysis of the state of the art and practical evaluation of Microsoft Hololens”, en, *arXiv:1706.08096 [cs]*, 2017. [Online]. Available: http://arxiv.org/abs/1706.08096 (visited on 11/28/2018).

[25] A. S. Carlin, H. G. Hoffman, and S. Weghorst, “Virtual reality and tactile augmentation in the treatment of spider phobia: A case report”, en, *Behaviour Research and Therapy*, vol. 35, no. 2, pp. 153–158, Feb. 1997. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S000579679600085X (visited on 12/03/2018).

[26] M. J. Schuemie, “Research on Presence in VR: A Survey”, en, p. 18, 2001.

[27] F. D. Rose, B. M. Brooks, and A. A. Rizzo, “Virtual Reality in Brain Damage Rehabilitation: Review”, en, *CyberPsychology & Behavior*, vol. 8, no. 3, pp. 241–262, Jun. 2005, ISSN: 1094-9313, 1557-8364. [Online]. Available: http://www.liebertpub.com/doi/10.1089/cpb.2005.8.241 (visited on 12/03/2018).

[28] N. Dt, “A randomized controlled study of a computer-generated three-dimensional model for teaching ear anatomy”, en, p. 22,

[29] V. A. Lahanas, “Augmented Reality for Simulation Based Training and Assessment in Minimal Invasive Surgery”, el, p. 257,

[30] Z. Xu, X. Lu, H. Guan, C. Chen, and A. Ren, “A virtual reality based fire training simulator with smoke hazard assessment capacity”, en, *Advances in Engineering Software*, vol. 68, pp. 1–8, Feb. 2014. [Online]. Available: `https://linkinghub.elsevier.com/retrieve/pii/S096599781300166X` (visited on 12/03/2018).

[31] A. Bellomo, M. Benassai, F. Lacquaniti, G. Mascetti, V. Maffei, W. L. Miller, and M. Zago, “VIRTUAL REALITY in NEUTRAL BUOYANCY (VRNB)”, en, p. 1,

[32] A. Gaitatzes, D. Christopoulos, and M. Roussou, “Reviving the past: Cultural Heritage meets Virtual Reality”, en, p. 7,

[33] S. Gonizzi Barsanti, G. Caruso, L. L. Micoli, M. Covarrubias Rodriguez, and G. Guidi, “3d Visualization of Cultural Heritage Artefacts with Virtual Reality devices”, en, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5/W7, pp. 165–172, 2015. [Online]. Available: `http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-5-W7/165/2015` (visited on 12/03/2018).

[34] R. J. K. Jacob, “INPUT DEVICES AND TECHNIQUES”, en, p. 38,

[35] P. J. Costello and P. Costello, “Health and Safety Issues associated with Virtual Reality - A Review of Current Literature”, en, p. 23,

[36] T. Johnsgard, “Fitts’ Law with a Virtual Reality Glove and a Mouse: Effects of Gain”, en, *Graphics Interface ’*, p. 8,

[37] L. Dipietro, A. Sabatini, and P. Dario, “A Survey of Glove-Based Systems and Their Applications”, en, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 461–482, Jul. 2008. [Online]. Available: `http://ieeexplore.ieee.org/document/4539650/` (visited on 11/30/2018).

[38] P. C. Grigore C. Burdea, *Virtual Reality Technology.* Wiley-IEEE Press, 2003. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=854CC6CFA1C610535A04CCA988D54

[39] S. V. Adamovich, G. G. Fluet, A. Mathai, Q. Qiu, J. Lewis, and A. S. Merians, "Design of a complex virtual reality simulation to train finger motion for persons with hemiparesis: A proof of concept study", en, *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 28, 2009. [Online]. Available: http://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-6-28 (visited on 11/30/2018).

[40] V. F. Pamplona, "The image-based data glove", en, p. 9,

[41] M. Hoppe, P. Knierim, T. Kosch, M. Funk, L. Futami, S. Schneegass, N. Henze, A. Schmidt, and T. Machulla, "VRHapticDrones: Providing Haptics in Virtual Reality through Quadcopters", en, p. 12,

[42] G. C. Burdea, "Haptic feedback for virtual reality", en, p. 11,

[43] T. Mazuryk and M. Gervautz, "History, Applications, Technology and Future", en, *VIRTUAL REALITY*, p. 72,

[44] J. Maletic, J. Leigh, A. Marcus, and G. Dunlap, "Visualizing object-oriented software in virtual reality", en, in *Proceedings 9th International Workshop on Program Comprehension. IWPC 2001*, Toronto, Ont., Canada: IEEE Comput. Soc, 2001, pp. 26–35, ISBN: 978-0-7695-1131-3. [Online]. Available: http://ieeexplore.ieee.org/document/921711/ (visited on 12/19/2018).

[45] *Cave*, https://i2.wp.com/www.techviz.net/wp-content/uploads/MWP0755.jpg, Accessed: 2019-01-13.

[46] T. H. Hollerer, "User Interfaces for Mobile Augmented Reality Systems", en, p. 238,

[47] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system", en, in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, San Francisco, CA,

USA: IEEE Comput. Soc, 1999, pp. 85–94, ISBN: 978-0-7695-0359-2. [Online]. Available: http://ieeexplore.ieee.org/document/803809/ (visited on 01/04/2019).

[48] O. Bimber and R. Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*, en. A K Peters/CRC Press, 2005. [Online]. Available: https://www.taylorfrancis.com/books/9781439864944.

[49] W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, J. Wieghardt, H. Hörtner, and C. Lindinger, "Augmented reality navigation systems", en, *Universal Access in the Information Society*, vol. 4, no. 3, pp. 177–187, 2006. [Online]. Available: http://link.springer.com/10.1007/s10209-005-0017-5 (visited on 11/28/2018).

[50] M. D. M. Noval, "Realidade Aumentada no ensino da Matemática: Um caso de estudo", pt, p. 132, 2013.

[51] G. Tholsgard, "3d rendering and interaction in an augmented reality mobile system", 2014.

[52] J. Glover, *Unity 2018 augmented reality projects: build four immersive and fur AR applications using ARKit, ARCore, and Vuforia*, en. 2018, OCLC: 1057891126, ISBN: 978-1-78883-876-4.

[53] P. Lang, A. Kusej, A. Pinz, and G. Brasseur, "Inertial tracking for mobile augmented reality", in *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276)*, vol. 2, May 2002, 1583–1587 vol.2.

[54] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hofmann-Wellenhof, "Global pose estimation using multi-sensor fusion for outdoor augmented reality", in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, Oct. 2009, pp. 153–162.

[55] R. Sood, *Pro Android Augmented Reality*, en. Berkeley, CA: Apress, 2012.

[56] *Pokemon go*, https://img.phonandroid.com/2016/07/pokemon-go-capture.jpg, Accessed: 2018-09-09.

[57] *Google glass*, https://analyticsindiamag.com/wp-content/uploads/2018/05/GoogleGlasses2_AGCO.jpg, Accessed: 2018-09-09.

[58] *Hololens*, https://compass-ssl.surface.com/assets/db/25/db253de9-73e2-4a2b-a209-7a3f01972f68.jpg?n=HoloLens_Homepage_Mosaic-Stryker_1920_890x400.jpg, Accessed: 2018-12-18.

[59] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays", en, in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, ACM Press, 1998, pp. 179–188. [Online]. Available: http://portal.acm.org/citation.cfm?doid=280814.280861 (visited on 12/18/2018).

[60] S. Mann, T. Furness, Y. Yuan, J. Iorio, and Z. Wang, "All reality: Virtual, augmented, mixed (x), mediated (x,y), and multimediated reality", Apr. 20, 2018. arXiv: 1804.08386. [Online]. Available: http://arxiv.org/abs/1804.08386 (visited on 02/21/2019).

[61] *Microsoft hololens*, https://cdn-images-1.medium.com/max/1600/1*Oltg1ajoJ1Xbs2fKON644g.jpeg, Accessed: 2019-03-26.

[62] *Magic leap one*, https://www.pcgamesn.com/wp-content/uploads/legacy/Magic_Leap_One_Creator_Edition-590x332.png, Accessed: 2019-03-26.

[63] S. Gualeni, *Virtual worlds as philosophical tools: how to philosophize with a digital hammer*. 2015, OCLC: 911495163. [Online]. Available: http://public.eblib.com/choice/publicfullrecord.aspx?p=4331932 (visited on 03/07/2019).

[64] *The mobile future of augmented reality 2018, qualcomm technologies, inc.* https://www.qualcomm.com/media/documents/files/the-mobile-future-of-augmented-reality.pdf, Accessed: 2019-03-26.

[65] *Beyond ar and vr - xr cross reality | cyrus lum*, https://www.youtube.com/watch?v=JM-Li92VSDo, Accessed: 2018-11-11.

[66] *Primeiro celular com reconhecimento de mundo da google é abandonado*, https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality, Accessed: 2018-11-11.

[67] *Bae systems leverages the power of mixed reality with microsoft and ptc*, https://www.youtube.com/watch?v=sDD-G32RqH8, Accessed: 2018-11-11.

[68] *Smart factory with microsoft hololens*, https://www.youtube.com/watch?v=6pyiiO72ZwM, Accessed: 2019-03-26.

[69] *Introducing dynamics 365 remote assist for hololens 2 and mobile devices*, https://www.youtube.com/watch?v=J-C6GE2gFYw, Accessed: 2019-03-26.

[70] *Industry partner solutions for microsoft hololens 2*, https://www.youtube.com/watch?v=uIHPPtPBgHk, Accessed: 2019-03-26.

[71] *Nba app for magic leap one | experience trailer*, https://www.youtube.com/watch?v=p7olw_LK_sI, Accessed: 2019-03-26.

[72] *Cnn app for magic leap one | experience trailer*, https://www.youtube.com/watch?v=8SicqW0_aXs, Accessed: 2019-03-26.

[73] *Ar-bril magic leap is veel beter dan de hololens*, https://www.youtube.com/watch?v=ujlRxJD_gCE&t, Accessed: 2019-03-26.

[74] L. M. D. Fernandes, “Guia de desenvolvimento de jogos para programadores independentes”, p. 110,

[75] *Unity 3d*, https://en.wikipedia.org/wiki/Unity_Technologies, Accessed: 2019-01-18.

[76] *Documentação inspect unity3d*, https://docs.unity3d.com/Manual/UsingTheInspector.html, Accessed: 2019-02-11.

[77] *Documentação prefab unity3d*, https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/prefabs-concept-usage, Accessed: 2019-02-11.

[78] “C# e Orientação a Objetos”, p. 202,

[79] *Documentação visual studio*, https://docs.microsoft.com/en-us/visualstudio/ide/?view=vs-2017, Accessed: 2019-02-11.

[80] *Documentação blender*, https://docs.blender.org/manual/en/latest/getting_started/about/history.html, Accessed: 2019-02-11.

[81] *Storyboard vr*, https://vimeo.com/193956977, Accessed: 2019-01-18.

[82] *Design inside virtual reality*, https://vrsketch.eu/, Accessed: 2019-01-18.

[83] *Blender render vs cycles*, https://cgcookie.com/articles/big-idea-blender-render-vs-cycles, Accessed: 2019-01-14.

[84] *18 ways to speed up blender cycles rendering*, https://www.youtube.com/watch?v=8gSyEpt4-60, Accessed: 2019-01-18.

[85] *Device class reference*, https://library.vuforia.com/content/vuforia-library/en/reference/cpp/classVuforia_1_1Device.html, Accessed: 2018-03-28.

[86] *Eyeweardevice class reference*, https://library.vuforia.com/content/vuforia-library/en/reference/cpp/classVuforia_1_1EyewearDevice.html, Accessed: 2018-11-11.

[87] *Renderingprimitives*, https://library.vuforia.com/content/vuforia-library/en/reference/cpp/classVuforia_1_1RenderingPrimitives.html, Accessed: 2018-11-11.

[88] *Targetfinder class reference*, https://library.vuforia.com/content/vuforia-library/en/reference/unity/classVuforia_1_1TargetFinder.html, Accessed: 2019-01-13.

[89] *Ivirtualbuttoneventhandler interface reference*, https://library.vuforia.com/content/vuforia-library/en/reference/unity/interfaceVuforia_1_1IVirtualButtonEventH html, Accessed: 2018-11-11.

[90] *Virtualbuttoncontroller*, https://bit.ly/2FAVap4, Accessed: 2018-11-11.

[91] *Objecttracker class reference*, https://library.vuforia.com/content/vuforia-library/en/reference/unity/classVuforia_1_1ObjectTracker.html, Accessed: 2019-01-13.

[92] *Videobackgroundmanager class reference*, https://library.vuforia.com/content/vuforia-library/en/reference/unity/classVuforia_1_1VideoBackgroundManager.html, Accessed: 2018-11-11.

[93] *Vr/ar/mr/xr: Where is the money?*, https://youtu.be/3IAdRrLpHEg, Accessed: 2019-03-14.

[94] *Focus mode*, https://library.vuforia.com/content/vuforia-library/en/reference/cpp/classVuforia_1_1CameraDevice.html, Accessed: 2018-11-11.

[95] *Synchro xr for the microsoft hololens*, https://www.youtube.com/watch?v=jdW7FiC4o2M, Accessed: 2018-11-11.