1 2 9 0

UNIVERSIDADE Ð
COIMBRA

Eduardo Rodrigues Catarino

Prototype of a Mixed Reality Videoconferencing Tool

Dissertation in the context of the Master in Informatics Engineering, specialization in Communication, Services and Infrastructures, advised by Professor Vasco Nuno Simões Pereira and Professor Jorge Carlos Santos Cardoso and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Đ
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Eduardo Rodrigues Catarino

Prototype of a Mixed Reality Videoconferencing Tool

Dissertation in the context of the Master in Informatics Engineering, specialization in Communication, Services and Infrastructures, advised by Prof. Vasco Nuno Sousa Simões Pereira and Prof. Jorge Carlos Santos Cardoso and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

Acknowledgements

The accomplishment of this work would not have been possible without the contribution of a group of people who have been beside me along the way. First and foremost, I would like thank my advisers Prof. Vasco Nuno Sousa Simões Pereira and Prof. Jorge Carlos Santos Cardoso for all their help and support.

Getting through my dissertation required not only academic support but also a great deal of friendship and companionship, I must thank all my colleagues and friends at the Department of Informatics Engineering, particularly Bruno Daniel Neto for all the help during the testing phase. This thesis also had the support of the GLN group who I must also thank.

Most importantly, none of this could have happened without my family, my deepest gratitude to my parents and my brother Tiago Filipe Rodrigues Catarino for their endless love and support.

Abstract

Mixed Reality (MR) merges real and virtual worlds, allowing the interaction between physical and digital objects in real time. While Virtual Reality (VR) creates artificial environments where users are completely immersed and Augmented Reality (AR) overlays digital information on the real world in a partially immersive experience, MR extends the concept of the latter by enabling a deeper interaction between real and virtual worlds. In a MR environment, real and digital elements interact between them and with the user, providing a more complete experience.

Video conferences allow for communication between several users remotely using real-time video and audio feeds. With this technology it is possible to hold meetings, lectures and presentations among as many people as necessary, from a one-on-one meeting to a lecture with hundreds of people. Users can participate in videoconferences with the use of computers, smartphones and other similar types of technology.

In this thesis, MR will be applied to a videoconference scenario to enrich the quality of experience of users in the same session. Using the platform created, local users can share their camera feed to present a real environment with artifacts that will be enhanced with a digital overlay. On the other end, remote users will receive the real-time feed from a real environment and can access the provided digital overlay with additional information. The digital overlay not only provides contextual information but also improves the interaction between the two type of users - local and remote.

Keywords

Mixed reality, Video conference, Interaction, Real-time video feed.

Resumo

Realidade Mista (RM) junta o mundo real e virtual, permitindo a interação entre objectos físicos e digitais em tempo real. Enquanto a Realidade Virtual (RV) cria ambientes artificiais onde os usuários estão completamente imersos e a Realidade Aumentada (RA) sobrepõe informações digitais no mundo real em uma experiência parcialmente imersiva, MR estende o conceito deste último, permitindo uma interação mais profunda entre o mundo real e virtual. Em um ambiente de MR, elementos reais e digitais interagem entre eles e com o usuário, proporcionando uma experiência mais completa.

Videoconferências permitem a comunicação entre vários utilizadores de forma remota utilizando transmissão de video e audio em tempo real. Com esta tecnologia é possível fazer reuniões, palestras e apresentações entre o numero de pessoas que for preciso, desde uma reunião um a um até uma palestra com centenas de pessoas. Os utilizadores podem participar em videoconferências com o uso de computadores, smartphones e outros tipos de tecnologia similares.

Nesta tese, a RM será aplicado a um cenário de videoconferência para enriquecer a qualidade da experiência de todos os utilizadores da mesma sessão. Usando a plataforma criada, os utilizadores locais podem compartilhar o seu feed de câmera para apresentar um ambiente real com artefatos que será aprimorado com uma sobreposição digital. Por outro lado, os utilizadores remotos iram receber o feed em tempo real de um ambiente real e podem aceder aos dados fornecidos pela sobreposição digital que têm informações adicionais. A sobreposição digital não só fornece informações contextuais, mas também melhora a interação entre os dois tipos de utilizadores - locais e remotos.

Palavras-Chave

Realidade Mista, Videoconferência, Interação, Feed de video em tempo real.

Contents

Acronyms

API Application Programming Interface.

AR Augmented Reality.

GPS Global Positioning System.

HDR High Dynamic Range.

HMD Head-Mounted Display.

ICE Interactive Connectivity Establishment.

IETF Internet Engineering Task Force.

IOT Internet of Things.

MR Mixed Reality.

NAT Network Address Translation.

OS Operating System.

OST Optical See-Through.

SDK Software Development Kit.

SDP Session Description Protocol.

SLAM Simultaneous Localization and Mapping.

STUN Session Traversal Utilities for NAT.

TURN Traversal Using Relays around NAT.

UI User Interface.

UWP Universal Windows Platform.

VR Virtual Reality.

VST Video See-Through.

W3C World Wide Web Consortium.

List of Figures

List of Tables

Chapter 1

Introduction

In today's world, technology is becoming more and more prevalent and is encroaching in almost every possible area of work one can think off. This is happening because of how technology can simplify or improve our day to day lives. In the workplace, innovative and creative ideas that can solve all sorts of problems are being developed, valued and incentivized. All companies need to be able to adapt and even reinvent themselves with new technologies in order to stay relevant and ahead of the competition. One of the technologies that is becoming increasingly more popular over time is the use of Mixed Reality (MR).

MR is the merging of real and virtual worlds to produce new environments and visualizations, where physical and digital objects co-exist and interact in real time. MR does not exclusively take place in either the physical world or virtual world, but is a hybrid of reality and virtual reality. This means that MR is closely related to the concepts of Augmented Reality (AR) and Virtual Reality (VR), the former focusing more in the physical world and the latter more in the virtual world [1]. As a matter of fact the development of such technologies started in the realms of AR and VR, and have over the years, with the improvement of technology, stared blurring the lines between the two creating the need for a new term which became MR.

VR can be traced all the way back to the 1800's with the creation of the stereoscope by Sir Charles Wheatstone [2]. Stereoscopes use mirrors to blend two images into one creating the illusion of depth. This was later improved on by Morton Heilig with the creation of the Sensorama Simulater in 1962. This simulator used stereoscopic experience and added stereo sound and aromas to increase immersion. The term VR became extremely popular during the 1980's with the invention of the data glove by Jaron Lanier. Since then the technology has made incredible strides with the creation of the VR headsets.

The term AR was first coined in 1990 by Thomas Caudell to describe the Head-Mounted Display (HMD) worn by Boeing's electricians [3]. This HMD would project a digital version of the wiring map onto plywood, instead of having the map physically printed into it. In 1992, Louis Rosenberg created the Virtual Fixtures which was one of the first fully functional AR systems. Ever since the early 2000's the technology has been used in many areas and types of work with the help of the new HMD and the increased use of mobile platform such as smartphones.

The term MR was first introduced in a paper called "A Taxonomy of Mixed Reality Visual Displays" in 1994 by Paul Milgram and Fumio Kishino [1]. The paper introduced the idea that MR is a continuum that includes VR and AR, as well as any new technologies that will allow digital content to interact with real world objects and humans to interact with digital objects as if they are real. As MR is becoming more accessible it is starting to be used in many areas such as manufacturing, aviation, education, entertainment, healthcare and remote work just to name a few.

1.1 Objectives

This dissertation project aims at uniting the technology of MR with the technology of videoconferencing. As technology that allow for communicating remotely is becoming more and more common and a necessity in today's world, videoconferencing is gaining an increased interest by many companies trying to adapt to the times. There is always a need for new and innovative ideas in the area, so this dissertation intends to combine the two technologies to create a interactive and immersive video conference experience using MR.

This work proposal has the support of the GLN group, a company engaged in the production both of technical molds of high precision and performance, and in highly technical level plastic injection. This dissertation will create a prototype of an application that can be used to make a real time visit to a GLN factory, with a presenter using videoconferencing to showcase (using MR) their molds and plastic injection machines to an online audience.

The work is going to include at first an in depth research and study into the concept of MR in order to understand all the necessary information for its use. Then a study on the existing tools that support the creation of MR application. This is going to be followed by the design of a generic platform to support the use of MR within a video conference. After this, a prototype of such a platform will be created. Finally this platform will be tested and validated.

1.2 Planning

The first part of the dissertation focused on the study and investigation on the topic of MR in preparation for the second part where the development of a prototype will happen. The plan consisted on research into MR, then of some tools that use it and finally of designing a generic platform to support the use of MR within a video conference.

The second part of the dissertation started with the study of possible videoconferencing tools that could support MR. This was immediately followed by the creation of the prototype where we put the two different chosen tools together (one for MR and the other for videoconferencing). After this, the development of the application continued with several other activities such as creating the required functionalities, deploying the prototype in a server and making design changes. In the end we had to test and validate everything in the prototype.

The plan was divided into fourteen different tasks:

1. Research on Mixed Reality - The topic of MR was studied and researched to better understand all of its possibilities and limitations as well as its history.

2. Research on Mixed Reality tools - After understating MR a research on existing tools that can help create a MR environment was done. This research involved the several features and applications of this tools.

3. Experimentation with Mixed Reality tools - After the research of the available tools it was important to experiment some of them to understand their features and possible limitations.

4. Definition of the Use Case - The Use Case of the project was specified and detailed in order to understand all the necessary requirements.

5. Design of a platform to support MR and Videoconferencing - Design of the initial application mockups considering all the requirements previously analysed.

6. Writing the interim report - The creation and documenting of everything that was done during the first part of the dissertation into a report.

7. Research on Videoconferencing tools - A big part of the intended application will involve the creation of a video conference. In order to do this several tools were researched.

8. Experimentation with Videoconferencing tools - After the research of the available tools it was important to experiment some of them to understand their features and possible limitations.

9. Research/set up of deployment scenario - With a videoconferencing application it will always be necessary to have servers running for people to connect to. For this reason we researched all available options and deployed the servers.

10. Integration of tools - It was necessary to integrate the chosen MR tool with the chosen Videoconferencing tool in such a way that they could both work as intended.

11. Design changes to the application - During the development of the application it was necessary to change it's User Interface (UI) so it could became more intuitive. This work includes the creation of mockups and the application of the UI updates.

12. Development of functionalities - This includes all the development of the features of the application that do not relate to the its MR part. It mostly includes the functionalities expected of a normal, commercialized videoconferencing application.

13. Testing and Validation - After creating the application it was required to make sure everything worked as expected. This process was spread out during a long time and was done after major tasks were completed. This includes the planning, documenting and performing of several tests.

14. Writing the Final Report - The creation and documenting of everything that was done during the first and second part of the dissertation into a report.

The dissertation plan is described in a Gantt graph for better visualization (Fig. 1.1). This graph also specifies the duration given for each task. Throughout the dissertation all tasks that are specified in the Gantt graph were completed. However their duration changed from the initial planning, several tasks took longer than expected as certain problems arose. With the extra time that was given to complete the project, we utilized it to improve the overall quality and have more in depth testing and validation.

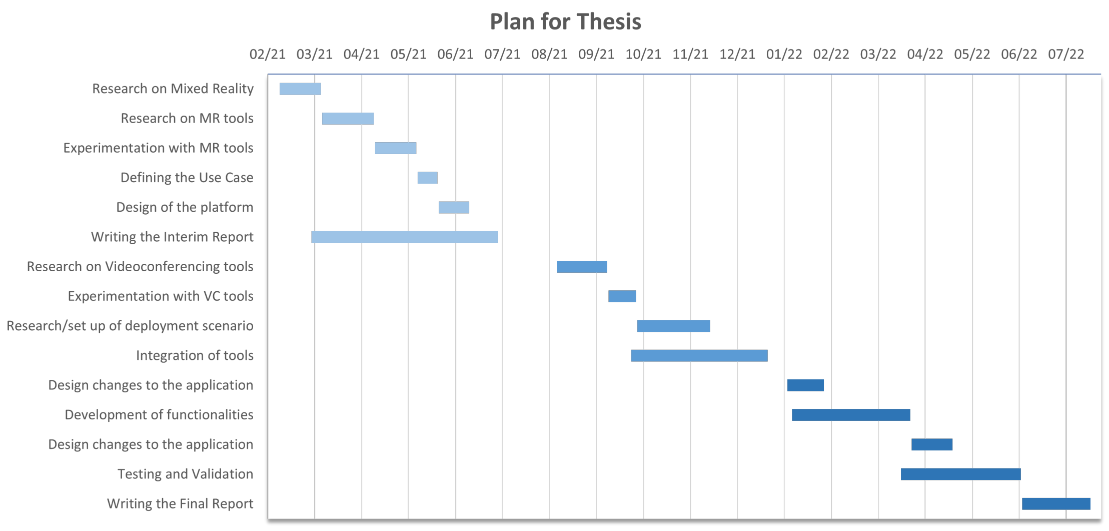


Figure 1.1: Gantt graph for the whole dissertation

1.3 Risks

In all projects there are certain risks associated to the development of any application. These can be unforeseen issues that affect the whole project. With this in mind, some of the possible risks that can happen in this specific project were listed and detailed in the tables bellow:

Name	Bad estimation on task duration
Description	While estimating the duration of a task it is possible to give it less time that what ends up being necessary
Consequences	Delay on other tasks
Probability	High
Impact	Medium
Mitigation Plan	Perform the tasks in order of importance so that less relevant tasks can be delayed or even not implemented

Table 1.1: Risk nº1

Name	Problems with the tools
Description	While working with the chosen tools for the project it is possible to encounter unforeseen issues, like an unexpected limitation of the tool
Consequences	The consequences can vary depending on the problem, development can be delayed and certain features can be scraped
Probability	Medium
Impact	High
Mitigation Plan	Have an in depth study of the tools to know what they can do and have other tools available for replacement if truly necessary

Table 1.2: Risk nº2

Name	Problems with integration
Description	The project requires to make an application that uses both Mixed Reality and Videoconferencing, the integration of the two tools might create issues
Consequences	The consequences can vary depending on the problem, development can be delayed and certain features can be scraped
Probability	Medium
Impact	High
Mitigation Plan	Have an in depth study and tests of the tools to know what can be done with them

Table 1.3: Risk nº3

Name	Failure to deploy server
Description	When trying to make the project accessible to many users, it is possible to not find a way to make it work
Consequences	Testing will become very difficult or even impossible
Probability	Low
Impact	High
Mitigation Plan	Attempt to always have a working solution even if only on a small private network

Table 1.4: Risk n°4

Name	Large bugs revealed during testing
Description	During testing it is possible to find project breaking bugs that must absolutely be fixed
Consequences	A possible delay to the project as the bugs must be solved and the project tested again
Probability	Medium
Impact	Medium
Mitigation Plan	During development ensure that central aspects of the project function, so that testing only finds small issues

Table 1.5: Risk n°5

During the development of this projects there were a couple of risks that came true. The first and most significant one was "Problems wit the tools" in table 1.2. What occurred was that a tool that had been researched and selected to be used in this project, was found to be inadequate for the use case after having already started to develop with it. As is stated in the mitigation plan, other tools were already researched and ready to take its place. This led to the consequences being only a delay in the development of the project.

The other risk that happened was the risk "Problem with integration" in table 1.3. Even after testing and knowing that the tools could work together, it was not until a bit later that a problem in the communication of MR elements to other users was found. Thankfully with the research done for the mitigation plan, this problem was resolved without changing tools but did result in a small delay.

1.4 Document Structure

This document is structured in 8 chapters in order to organize its contents. The chapters are divided as follows:

Chapter 1 - Introduction

A description of the dissertation is made. The planning of the semester is shown as are the risks involved in developing this project and the structure of the document.

Chapter 2 - Background

In this chapter the definition of MR is shown, alongside the several types of options that MR can have, such as the types of tracking and displays.

Chapter 3 - State-of-the-art

In this chapter there is a description of the many tools that were researched in the topics of MR and videoconferencing with a critical analysis of the tools.

Chapter 4 - Experimentation

This chapter has a detailed description of the tests that were executed in order to better understand the tools that were researched. Only the tools that were seen as the most useful in the previous chapter were experimented.

Chapter 5 - Requirements Analysis

In this chapter a use case is described for this project with its requirements listed with all design and functional choices explained.

Chapter 6 - Architecture & Implementation

This chapter describes all the work done in developing the application. It is in this chapter that all the choices that were made for this project are summarized. It also explains its structure and architecture.

Chapter 7 - Evaluation

In this chapter all the tests and evaluations done to the project are explained together with all the results and conclusions that were made.

Chapter 8 - Conclusion

A reflection of what was done in this project and an analysis of what is to come next is made.

Chapter 2

Background

This chapter describes the main concepts of MR while also listing and explaining the different approaches that already exist in order to create a MR Environment. Initially, the concepts of AR and VR are defined in order to distinguish them from MR because of how common they are in this area and because their definition helps to better understand the concept of MR. Then the idea of what MR is and how it differs from other types of reality is explained. After that, from a theoretical standpoint, several tracking options that have been used with MR are explored. Finally, multiple types of displays that can be used in MR scenarios are examined.

2.1 Augmented Reality

AR can usually be defined by the addition of virtual information on top of the real world in real time (while the term AR has appeared in many different works its definition has not always been consistent) [1]. The name comes from the idea of having a real environment and augmenting/enhancing it with virtual information (this information can be objects, audio and other sense enhancements [4]). In AR, as is to be expected, it is necessary for the user to be tracked in terms of location and orientation. Describing AR is also frequently done with the concept of interaction, the experience that allows the user to engage with the augmented physical world in a satisfying manner.

A possible scenario of where AR can be used is for instance in the purchase of a new house. This house is completely empty with no furniture and the buyer wants to know what the house would look like after it has been furnished. Now let us consider an AR experience that allows the buyer to place whatever objects they want wherever they want, they can also walk all over the house and view it from whatever angle they desire (Fig. 2.1). AR can also offer the possibility to do things that would not be possible in the real world. An example is the now popular game Pokemon GO. In it a user can use their smartphone to see the Pokemon added into the real world which adds immersion into the game (Fig. 2.2).

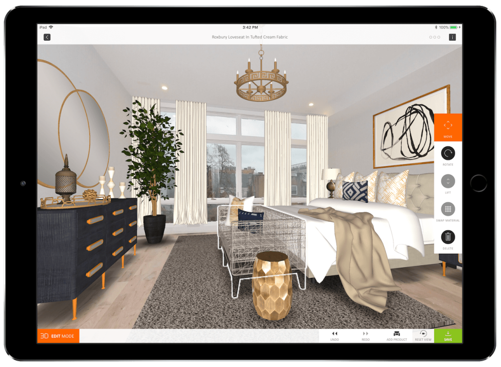


Figure 2.1: Example of a room being enhanced with AR [5].

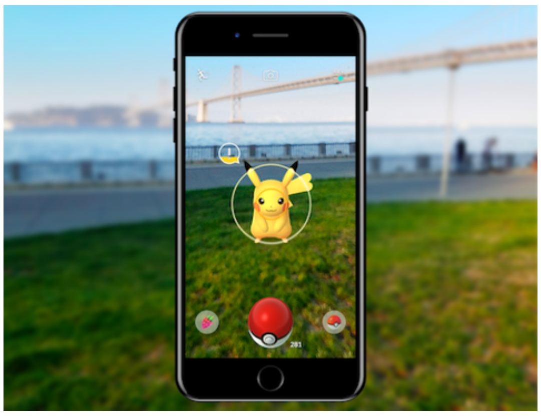


Figure 2.2: Example of a game being enhanced with AR [6].

2.2 Virtual Reality

VR is when the user is completely immersed in a virtual world, whether it represents or functions as the real world or a fictional one. It is a medium composed of interactive computer simulations that sense the participant's position and replace or augment the feedback to one or more senses, giving the feeling of being immersed or present in the simulation [7]. It is important to emphasise the fact that in VR the environment is fully synthetic and that the VR system tracks your location and orientation but is not fixed to a specific location of the real world.

VR can be used for example in aviation, medicine and in the military, as a training tool. With VR there is no need to train with expensive equipment, in dangerous situations or with sensitive technology [8]. Pilots can train in realistic but virtual cockpits, surgeons can train with virtual patients and the military can conduct virtual raids without putting themselves at risk (Fig. 2.3). Another good example is the use of VR to treat mental illness [9]. Post-traumatic stress disorder and other serious conditions can benefit from therapy sessions that use VR technology (Fig. 2.4).

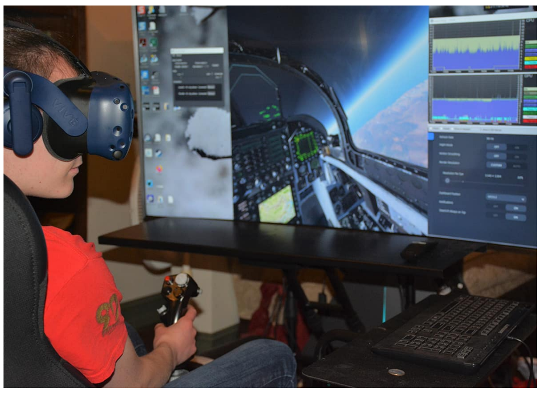


Figure 2.3: Example of flight simulation using VR [8].

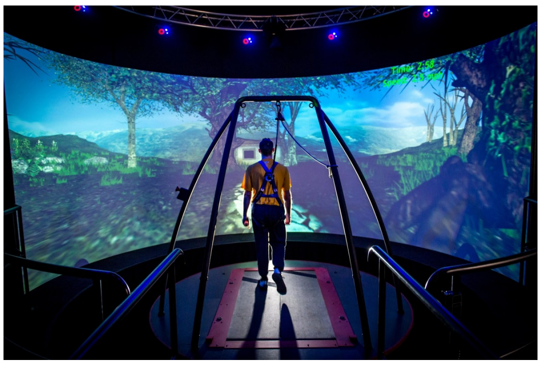


Figure 2.4: Example of a therapy session using VR [9].

2.3 Mixed Reality

While the two concepts of AR and VR can seem quite different it is important to know that they are related and can be considered together. The most convenient way of doing this is to consider them inside of a continuum, and it is with this continuum that we can more easily define and understand MR.

In Milgram's Reality-Virtuality Continuum [1] it is explained that there is a continuum that depicts the possibilities of the interaction between the real world and the virtual world. In this continuum, at the far left side we have the real environment, where there are only real objects, and at the right side we have the virtual environment, in which there are only virtual objects. The MR environment, the subject of this thesis, is anywhere between this two extremes, one where there is both a real and a virtual environment. AR, in the continuum, is located closer to the real environment, while VR is at the right most extreme in the virtual environment. Closer to the virtual environment we have Augmented Virtuality which introduces real objects in a virtual environment. For example when it is possible to visualize a real time model of a component of a larger machine from a computer screen, this is considered Augmented Virtuality because a real object (the component) is being seen in a virtual environment (the computer screen).

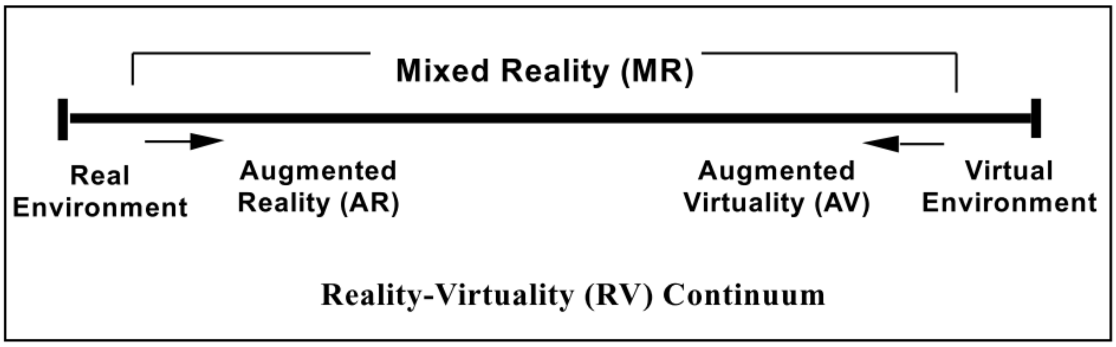


Figure 2.5: Simplified representation of a RV Continuum [1].

MR can be best defined as the overlaying of virtual objects onto the real world [10] in a way that allow interactions between the two. More often than not this definition is thought of when thinking about AR. In truth the term AR has began to be used in several different scenarios and it can be hard to understand what people mean when they use it [11]. The two concepts usually overlap because AR can be considered to be contained within MR, as can VR be. It is therefore possible to say that all AR and VR applications are considered MR but that the opposite is not true [11], MR is a much more broad concept that encompasses many more types of applications that involve the interaction of the real and virtual worlds.

An example of a type of application that is considered as MR but not any of the other two is the use of a real world object to interact with a digital world, using that object in the way it is used in the real world (Fig. 2.6). Another example is for instance the Global Positioning System (GPS) devices which use GPS information to indicate location and direction (via a digital information) on a map (which represents the real world). While there is digital information with spatial awareness being added to a representation of the real world it is not considered AR but is MR (Fig. 2.7).

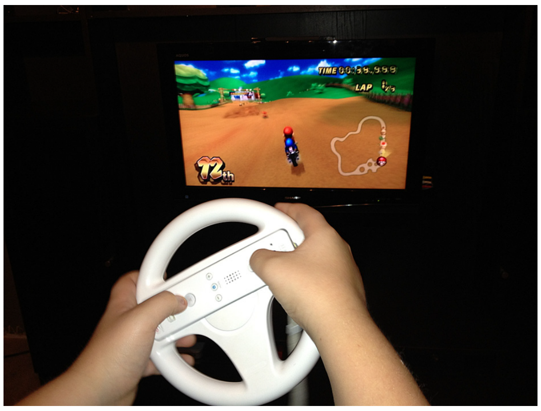


Figure 2.6: Example using a real world steering wheel to interact with the Wii system (digital world) [11].

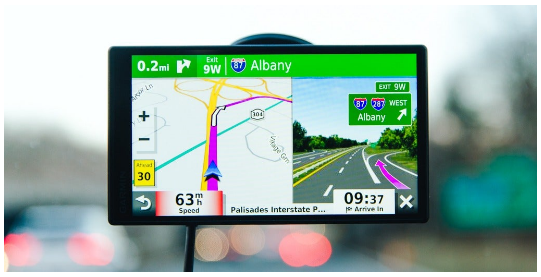


Figure 2.7: Example of a GPS device [12].

2.4 Tracking

Tracking has been one of the most popular topics of research in the area, most likely because of its importance in enabling technologies for MR [13]. Real time tracking is a problem that does not have a single solution, there are many different approaches that can be taken to solve it. Multiple tracking options already exist in today's market and most of them are being researched for many different types of scenarios. Tracking in itself is the process of determining the position of a user or an object in an environment in the most accurate way possible [14]. It is important to note that position here refers to both location and orientation. When it comes to tracking the user, more often than not this is done with the sensors that come with the used hardware, normally smartphones or HMD (Fig. 2.8). For object tracking there are many used methods that make it possible to determine where the object is(Fig. 2.9).

Figure 2.8: Example of a HMD that can track the user (Microsoft Hololens 2) [15].

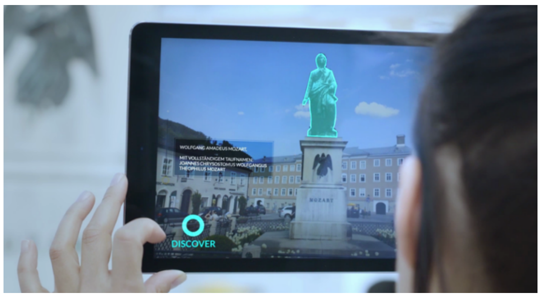


Figure 2.9: Example of a statue being tracked [16].

One of the most common techniques is vision based tracking that can be divided in two different types, markerless tracking and marker tracking. Markerless tracking is based on tracking through natural features already available in the scene, and so the system needs to find these features in the video feed [17] (e.g. detecting flat surfaces in order to add virtual objects on top)(Fig. 2.10). Much more commonly used is marker tracking in which markers are added to the scene in order to facilitate tracking. Markers (or fiducial markers) are distinguishable elements put in the environment so that they can be easily identified apart from other objects [14]. They can be categorized into active markers and passive markers. Active markers emit some sort of signal that can be captured by a sensor (e.g. magnetic, light) while passive markers are typically patterns that can easily be isolated (e.g. QR codes)(Fig. 2.11).

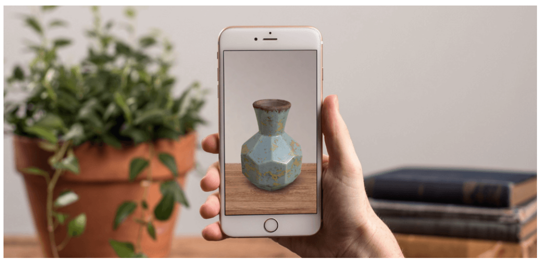


Figure 2.10: Example of surface being tracked and an object being placed on top (markerless tracking) [18].

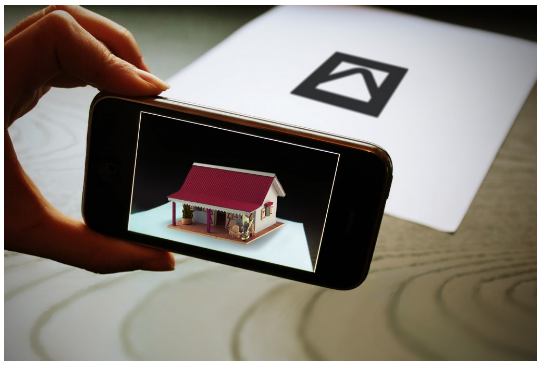


Figure 2.11: Example of symbol being tracked and an object being placed on top (marker tracking, passive marker) [19].

Tracking can be achieved with two methods, outside-in, in which the markers are mounted on the user and the sensors are mounted in the environment (usually involves the user using a hat-like item with several fiducial markers), and inside-out, in which the sensors are mounted on the user and the markers are mounted around the environment (the sensor the user is carrying can be for example a HMD or a smartphone).

It is also important to note that there are tracking techniques that use the aforementioned sensors, such as magnetic and light, without the vision part. This sensors and many others like acoustic, inertial or optical, just to name a few, can be used for tracking in some way, either by themselves or in conjunction with other methods, all with their own advantages and disadvantages [13]. This tracking technique that uses several different sensors is called Simultaneous Localization and Mapping (SLAM)(Fig. 2.12). There also exists many GPS based tracking solutions used for outdoor tracking, usually combined with inertial sensors such as accelerometers and gyroscopes(Fig. 2.13).

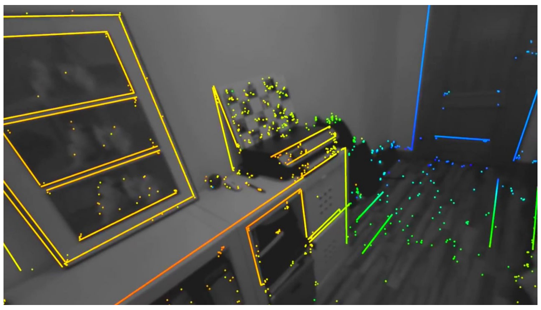


Figure 2.12: Example of an area being tracked with SLAM technology (this is the map that results from the information given from the several sensors) [20].

Figure 2.13: Example of a street with digital information from GPS tracking [21].

Many of this methods and sensors could be used to provide a tracking solution, but several new techniques use a fusion (hybrid) approach. These approaches try to use many available techniques in conjunction in order to provide robust solutions, for example using a tracking system based on GPS, inertial and computer vision sensing [13]. All of the aforementioned tracking methods can be summarized in Fig. 2.14.

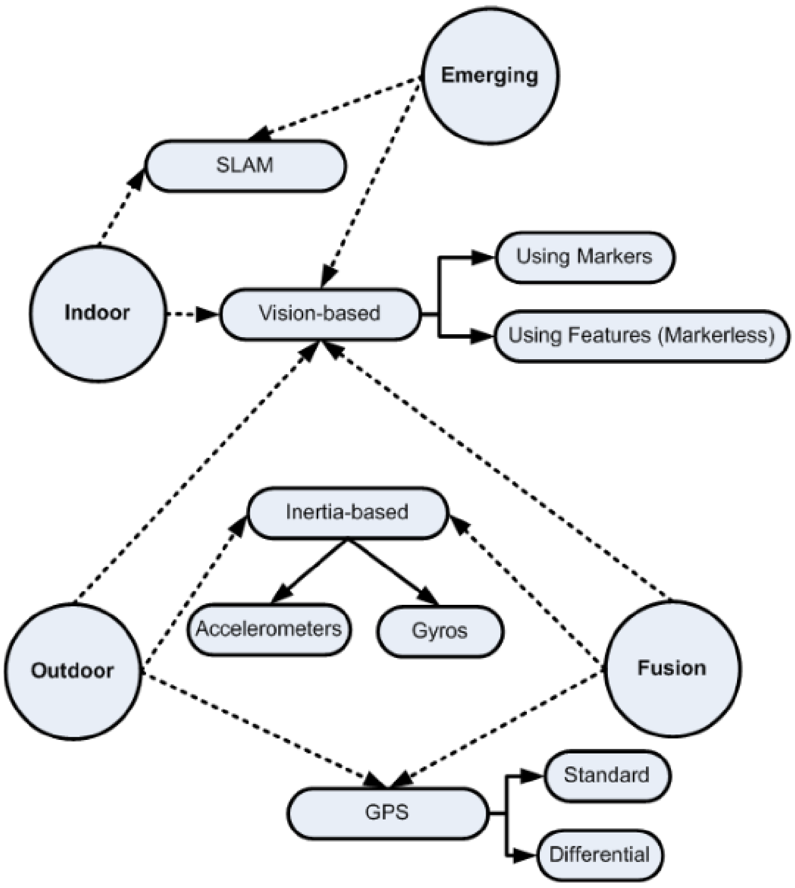


Figure 2.14: Tracking Methods [14].

2.5 Displays

The display is the device that provides the signals that our senses perceive [11]. There are a few display types that are frequently used in MR scenarios, these displays are desktop computing displays, HMD, projection-based displays and handheld displays.

Desktop computing displays are simply using a desktop computer and a screen to display the MR environment with the possibility of interaction with a mouse, for example.

HMDs can be divided in two types, non-see-through HMD or see-through HMD. Non-see-through HMD (such as VR headsets) are HMD that put screens in front of the users eyes in order to simulate a virtual environment with the intention to immerse the user in that environment (Fig. 2.15). The video or image placed on the VR screen is split in two, with an individual view for each eye to create a 3D perspective. All VR screens will also utilize lenses between the screen and your eyes. This helps distort the screen presented image into something more lifelike for our eyes. Non-see-through HMD can range from simple cases made from cardboard and a few lenses to being filled with sensors and other technology. The most popular options are PC-based like the HTC Vive, Oculus Rift and Playstation VR. There are also smartphone options such as Samsung Gear VR and Google Cardboard. There is a type of VR headset that does not need a platform (like PC or a smartphone) to support it, they are standalone VR headsets, some good examples are the Oculus Go and the Daydream headset [22].

See-through HMD also come in two types, Optical See-Through (OST) and Video See-Through (VST) [13]. OST displays are those that permit the user to see the real world with their eyes and then overlay graphics onto the users view. In OST HMD the direct view of the world is mostly preserved and there is no perspective conversion in viewpoint and field of view, this allows the user to maintain an unaltered and almost natural visual experience of the surrounding world (Fig. 2.16). Nowadays OST HMD are at the cutting edge of the AR research, and several consumer level headsets have been recently developed following the success of the Microsoft HoloLens 1 (e.g., MagicLeap One, HoloLens 2, Meta Two, Avegant, Lumus DK Vision) [23].

VST displays are those in which the user sees a video of the real world with graphics overlaid on it. On VST systems the real-world is captured by one or multiple cameras, normally located in front of the HMD. VST systems are not very common in today's market for their lack of popularity, probably because of the depth perception issues that this type of HMD has.

Figure 2.15: Example of a VR Headset [24].

Figure 2.16: Example of a OST HMD [25].

Projection-based displays are used to display graphical information directly on real objects or surfaces, providing minimal intrusiveness [13]. The projected images can be computer generated or photographic, and either pre-rendered or generated in real time. Projection-based AR typically uses one or more projectors arranged around an object (such as a prop or character in an attraction) or distributed throughout a 3D space (such as an entire scene). If the display uses multiple projectors, their images can be independent of one another or blended together, either manually or via automatic techniques. These displays can casually align images with physical objects or precisely register them to align with surface features. If precisely registered, the projected images can replicate surface colors and features to create a multiplicative effect on color, saturation, and contrast. Doing so yields stunning High Dynamic Range (HDR) results. The main advantage of projection-based AR is that it can create beautiful dynamic environments and bring sets to life in a magical way difficult to achieve with traditional lighting. HDR lighting, per-pixel control, animated media, and interactive content are all exciting new tools for use by theme park designers [26].

There are several types of handheld displays available to the common user, such as tablet PCs and smartphones. The easy access to a handheld display, its high mobility and its minimal intrusiveness makes this type of display a great alternative to the popular but harder to get HMD. Luckily nowadays better and better handheld displays are becoming available to the normal consumer for lower prices, and since the technology regarding MR has been around for a while it is available for many different platforms. The most commonly used Operating System (OS) are Android, iOS and Windows. Tools used to develop MR applications support one or more of this OS with a list of recommended devices that can be used per OS.

Chapter 3

State-of-the-art

This chapter looks at all the tools available for the creation of MR applications. The tools that were looked at were libraries or toolkits that introduce MR elements and libraries or platforms that enable videoconferencing in an application. This analysis was made with the intention of choosing the best tools to be used in this thesis. The chapter starts by making an analysis of available MR tools starting with several simpler tools that were studied in order to start understanding how they work. These tools were chosen for their popularity or having specific uses that could be considered for our project. The last two tools studied (Wikitude and Vuforia) are complex tools with many features, for this reason they were given a much deeper look. Finally we looked into tools available to help create the videoconferencing part of the project. Only tools that had free versions were researched and analysed.

3.1 Available Mixed Reality Tools

The realm of MR has become more and more popular over the last decade and because of this a large number of tools to help developers create application that involve MR have been made. Several of this tools were researched for this paper while considering the features they enable (for example what type of tracking they allow), how they work, their history and what type of use cases they can be used for. It is also important to note that this study was done with the scenario of a video conference in a factory in mind, therefor research was focused on tools with potential for this case.

3.1.1 ARKit

ARKit is a package of profiled tools first introduced during the WWDC 2017 event. It offers a mostly new approach to AR technology. In particular, the solution is capable of identifying dimensions of the surrounding and consider lighting conditions to integrate virtual objects with real life as seamlessly as possible. This toolkit for AR development was created and launched specifically for the iOS platform by Apple. This means that it is possible to create apps only for iPhones and iPads with this package, so it may be a good AR toolkit if you aim to make an AR exclusively for these devices [27].

ARKit comes with the Depth Application Programming Interface (API) which allows for the scanning of scenes in order to place augmented content in the scene and blend them seamlessly with their physical surroundings. ARKit also has instant tracking which recognizes horizontal surfaces to augment, and it also includes face tracking support of up to 3 people with the objective of adding some sort of virtual model on top of ones face [28].

3.1.2 ARCore

ARCore is quite similar to Apple's ARKit but made by Google. Unlike ARKit, ARCore allows anyone to develop AR apps for both Android and iOS. Using different APIs, ARCore enables your phone to sense its environment, understand the world and interact with information. Some of the APIs are available across Android and iOS to enable shared AR experiences.

ARCore uses three key capabilities to integrate virtual content with the real world as seen through the phone's camera: Motion tracking which allows the phone to understand and track its position relative to the world. Environmental tracking (with instant tracking) that allows the phone to detect the size and location of all type of surfaces such as horizontal, vertical, angled like the ground, a coffee table or walls so that it can attach augmented objects to its surfaces. Light estimation allows the phone to estimate the environment's current lighting conditions in order to react to the changes in the lighting volume and even generate shades [29].

3.1.3 ARToolKit

ARToolKit is a package of open-source libraries for developing AR apps based on marker recognition. The main principle of ARToolKit's performance is that it processes graphic data received from the mobile camera in the runtime mode guided by the initially known square object markers. After that, the objects are displayed on the smartphone screen with the consideration of their location in space. With ARToolKit, it is possible to create solutions for iOS, Android and Windows [27].

One of the key difficulties in developing AR applications is the problem of tracking the users viewpoint. In order to know from what viewpoint to draw the virtual imagery, the application needs to know where the user is looking in the real world. ARToolKit uses computer vision algorithms to solve this problem. The ARToolKit video tracking libraries calculate the real camera position and orientation relative to physical markers in real time. This enables the easy development of a wide range of AR applications. This also means that ARToolKit only has vision based, marker tracking [30].

There are some limitations to purely computer vision based AR systems. Naturally the virtual objects will only appear when the tracking marks are in view. This may limit the size or movement of the virtual objects. It also means that if users cover up part of the pattern with their hands or other objects the virtual object will disappear. There are also range issues. The larger the physical pattern, the further away the pattern can be detected. This range is also affected somewhat by pattern complexity. The simpler the pattern the better, but there is a limit to how simple it can be. If the pattern is too simple the ARToolKit may not be able to recognize it as a marker. Tracking is also affected by the marker orientation relative to the camera. As the markers become more tilted and horizontal, less and less of the center patterns are visible and so the recognition becomes more unreliable. Finally, the tracking results are also affected by lighting conditions. Overhead lights may create reflections and glare spots on a paper marker and so make it more difficult to find the marker square. To reduce the glare, patterns can be made from more non-reflective material. For example, by gluing black velvet fabric to a white base. The 'fuzzy' velvet paper available at craft shops also works very well [31].

ARToolKit is made available under a dual-license model. As it has been since the first public release of version 1.0, it is freely available for non-commercial use under the terms of the GNU General Public License. A proprietary version of ARToolKit developed in parallel for professional commercial use by ARToolworks Inc. is made available under different license terms, to suit end-user needs. These license arrangements are managed by the company. A variety of license types are available at reasonable cost. It is necessary to contact ARToolworks Inc. directly for information on products, license terms, and pricing [32].

3.1.4 AR.js

AR.js is a lightweight library for AR on the Web, coming with features like Image tracking, Marker tracking and Location based AR. Image tracking allows a 2D image to be found by the camera in order to add some sort of content on top of it, Marker tracking allows for the same as Image tracking but with specific markers that are limited in shape, color and size. Location based AR uses real world places as markers in order to show augmented content.

The best features of AR.js are the fact that it runs very efficiently even on phones, that no installation is required since it is a pure web solution (full javascript based on three.js + A-Frame + jsartoolkit5), that it is open source, free and that it works on any phone with webgl and webrtc.

For developers, to develop AR on the Web, means to avoid all the Mobile app development efforts and costs related to App stores (validation, time to publish). It also means to re-use well known technologies like Javascript, HTML and CSS, known from a lot of developers and possibly designers. It basically means that is possible to release every new version instantly, fix bugs or release new features in near real-time, opening a lot of practical possibilities.

For users, it means to reach an AR experience just visiting a website. As QR Codes are now widespread, it's also possible to scan a QR Code and reach the URL without even typing. Additionally, users do not have to reserve storage space on their phone to download an AR app, and do not have to keep it updated [33].

3.1.5 Wikitude

The Wikitude AR Software Development Kit (SDK) is a very popular framework for creating AR-based apps with a community of over 150,000 registered developers and has powered more than 40,000 apps across multiple industries. Wikitude is a dynamic AR technology company that has existed since 2008. This SDK is constantly being improved upon and updated [34].

Wikitude allows for the use of many different types of tracking, while boasting cross-platform support and a broad framework choice. Using Wikitude it is possible to create projects for many platforms such as Android, iOS, Windows and smart glasses allowing for AR experiences in smartphones, tables and is optimized for certain smart glasses such as Epson Meverio, HoloLens and Vuzix. Development in Wikitude can be done in several development environments like JavaScript, Unity, Cordova, Xamarin, Flutter and Native API [35]. With the use of all this platforms and development enviroments being well documented at https://www.wikitude.com/documentation/.

When it comes to tracking, Wikitude has Image tracking that lets AR apps detect, track and augment 2D images, while also having multi Image tracking which allows for several images to be tracked simultaneously while arranged in an arbitrary way or into regular geometric shapes like cubes. Similar to Image tracking, Wikitude has cylinder tracking which has 2D images wrapped onto objects that are cylindrical as is the example of soda cans. Another type of tracking is object tracking that enables real-time 360º AR experiences around physical objects such as toys or machines, and also supports multi objects tracking which allows for several objects to be tracked simultaneously. Wikitude is also able to track environment making it possible to recognize, track and augment feature-rich rooms and scenes (making it possible to augment a whole room full of virtual objects). It also has instant tracking that enables the overlay of interactive digital augmentations to physical surfaces (such as adding a virtual vase on top of a table), this particular tracking in Wikitude also allows it to be dynamically connected to ARKit and ARCore making it possible to interchange between them giving an API with greater reach and cross platform (Wikitude SMART). The last type of tracking given by Wikitude is Geo AR which boosts geographical points of interest with meaningful content using GPS [35], this way it is possible for example to add relevant information to a users surroundings (the types of tracking supported are summarized in table 3.1).

In order to use the Wikitude SDK one must first chose a package to use. The "Enterprise" package can have all the available features in the SDK but the details must first be discussed with the company in order to chose all the necessary features and to arrive at a price. There is also the "PRO 3D" package that does not have the multi object and cylinder tracking while also not having production support and access to development of applications for smart glasses. This package has 2 payment types, the first is a one time fee of 2490/€ but no updates and the second is a yearly subscription of 2990/€ with updates. Another package is the "CLOUD" which is the same package as the "PRO 3D" but with cloud hosting (can consist of up to 50,000 images) coming at a yearly subscription of 4490/€. The final package is the "Free trial" that comes with all the features in Wikitude but has a limit use of 45 days, includes a watermark and gives no publication rights [36]. The information on the relevant packages for this project is summarized in table 3.1.

3.1.6 Vuforia

The Vuforia Engine is one of the most widely used platform for AR development, with support for smartphones, tablets, and eyewear. Developers can easily add advanced computer vision functionality to Android, iOS, and Universal Windows Platform (UWP) apps, to create AR experiences that realistically interact with objects and the environment [37]. PTC Inc. (formerly Parametric Technology Corporation), the owners of Vuforia engine are a computer software and services company founded in 1985. PTC products and services include Internet of Things (IOT), AR, and collaboration software [38].

Vuforia engine boast over 800,000 developers worldwide making it one of the most popular AR software in the market. It allows for a great variety of use case scenarios to be created from its use, with many different types of tracking. Vuforia also provides several different tools used specifically to support the development with the engine. The engine also makes it possible to create an application for many different platforms and allows for the development in several different environments [39].

Tracking in Vuforia engine comes in many forms. It has Image tracking which allows for previously known images to be used as targets. This includes normal 2D images but also cylinder targets, which are images wrapped onto objects that are cylindrical or close in shape (e.g. beverage bottles, soda cans), multi-targets, which are several image targets arranged into any arbitrary arrangement of planar surfaces or into regular geometric shapes (e.e. cubes), and VuMarks that are customizable markers that can encode a range of data formats. Vuforia engine also allows the tracking of 3D objects with 2 different methods. The first is called model targets and it allows the engine to recognize objects by using pre-existing 3D models and tracking its shapes, this is preferable for big objects like industrial equipment, vehicles and home appliances. The second method is called object targets which are created by scanning an object with the Vuforia Object Scanner (one of the tools available to support development mentioned earlier). This is the best option for detecting and tracking intricate 3D objects such as toys. The engine also permits the tracking of several objects at the same time (multi objects tracking). Another type of tracking used in Vuforia is environment tracking which is done by scanning a limited area and then allowing the user to add persistent content to it. Finally the last type of tracking is instant tracking where the engine can recognize horizontal surfaces and then add content on top (the types of tracking supported are summarized in table 3.1) [40].

As mentioned before, Vuforia has several tool at the developers disposal in order to help them with their project. A few examples of such tools are the "Vuforia Object Scanner" (it helps easily scan 3D objects into an object file to upload to Vuforia), the "Model Target Generator" (a desktop application that allows for the creation of model targets from pre-existing 3D models), the "Model Target Test App" (assist in evaluating model targets), the "Vuforia VuMark Designer" (it allows for the creation of custom VuMarks), among several others [41].

In terms of developments this engine allows for the creation of projects in Android, iOS and UWP, with support for smartphones, tablets and many AR headsets and glasses such as the Microsoft HoloLens 2 and RealWear HMT-1 for example. The development can be done using several different environments such as Unity, Visual Studio, Android Studio and Xcode [42]. Vuforia Engine also provides a detailed documentation with several examples on the varied platforms and development environments found at https://library.vuforia.com/.

When it comes to licensing, in order to use Vuforia Engine one of several packages must be selected. The most complete package that includes all available features and a customizable cloud database usage is the "Pro" package which requires the contact of the company in order to negotiate a price. There is also the "Basic" and the "Basic+Cloud" packages which do not allow for the use of model targets, area targets, advanced APIs and production support with the "Basic" package also not allowing cloud storage usage while the "Basic+Cloud" allows cloud usage with only using 10000 recos per month and 100000 images (A "reco" occurs when your app recognizes a target in your Cloud Database [43]). These packages come at the price of 42$ per month for the "Basic" and 99$ per month for the "Basic+Cloud". There are also the "Lab Pack" and "University Pack" which come with 50 and 500 licenses respectively with all features except for production support and the "University Pack" doesnot allow Cloud usage. For these packages the price also needs to be arranged with the company. Another package is the "Agency Package" that comes with 5 short-term licenses with the possibility of all features or only some, with the pricing being also arranged with the company. The last package is the "Developer" which is free to use but it does not allow for the publishing of the project for commercial use, it also does not have production support and has a few limitation and a watermark. The limitations are that the developer is only allowed 100 VuMarks, 20 model targets, 10 area targets and the cloud database is limited to 1000 recos per month and 1000 images [44]. The information on the relevant packages for this project is summarized in Table 3.1.

3.2 Comparing Wikitude/Vuforia

Wikitude and Vuforia are two similar and very large tools with many features and several packages to chose from. It would therefore be important to compare the two in order to chose one, as experimenting with both would take a substantial amount of time. This way we experiment with one or more of the smaller tools and only one of the large ones. With that in mind a table was created in order to compare the two tools in terms of packages, available platforms, tracking methods and other relevant criteria (table 3.1).

		Vuforia				Wikitude			
Products	Packages	Developer	Pro	Basic	Basic+cloud	Free trial	Enterprise	Pro 3D	Cloud
	Price	Free*	Custom	504$ year	1188$ year	Free*	Custom	2990€ year	4490€ year
Platforms	Android	x	x	x	x	x	x	x	x
	iOS	x	x	x	x	x	x	x	x
	UWP	x	x	x	x	x	x	x	x
Tracking	Image	x	x	x	x	x	x	x	x
	Multi Image	x	x	x	x	x	x	x	x
	Object	x	x	x	x	x	x	x	x
	Multi Object	x	x	x	x	x	x		
	Cylinder	x	x	x	x	x	x		
	Model	x	x						
	Environment	x	x			x	x	x	x
	Instant	x	x	x	x	x	x	x	x
	VuMarks	x	x	x	x				
	Geo AR					x	x	x	x
Cloud*		x	x		x	x	x		x
Watermark		x				x			

Table 3.1: Comparison Wikitude/Vuforia

*(Has some limitations; these limitations are described in the texts regarding the tools, section 3.1.6 and section 3.1.5)

As can be seen from table 3.1 both tools give access to the most common and useful features one can have in MR, the exceptions being that Vuforia provides model tracking and VuMarks and Wikitude provides Geo AR. Model tracking is used for tracking big objects like industrial equipment and vehicles, which given our case can be an interesting choice and VuMarks are very customizable and allow for the transmission of many data formats making it also a feature worth looking at. Geo AR used in Wikitude makes use of geographical points and meaningful content using GPS so it's used mostly for outdoor scenarios, which in our case is not applicable. So Vuforia comes on top with it's choice of features, but what really makes Vuforia a more sensible choice than Wikitude is the fact that Wikitude's free version only has a 45 day trial while Vuforia's free version doesn´t have a time limit.

3.3 Available Videoconferencing Tools

The main objective of this project involves the creation and handling of a video conference application on which we will add MR elements. It is therefore important to also study and analyse the existing tools to create video conference applications. Note that only freely available tools were considered for this study.

3.3.1 WebRTC

WebRTC [45] is a free and open-source project used to add real time communication capabilities to web browsers and application that work on top of an open standard via an API. It allows for video, voice and generic data communication between peers (P2P) without the need to install plugins or download native apps. The technology is available on all modern browsers as well as on native clients for all major platforms (it is supported by Apple, Google, Microsoft and Mozilla, amongst others).

In May 2011, WebRTC was released by Google as an open-source project for browser-based real-time communication. WebRTC specifications have been published by the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF). Since then this has been followed by ongoing work to standardize the relevant protocols in the IETF and browser APIs in the W3C. WebRTC is a popular and well liked basis for many videoconferencing application out there [46].

WebRTC uses three different APIs. The WebRTC API getUserMedia is built into Chrome and Firefox and allows browsers to capture the output from the camera and microphone to stream voice and video. The RTCPeerConnection API connects your local computer with the remote peer and maintains a stable connection for efficient communication. Finally, the RTCDataChannel API enables the exchange of all non-audiovisual data (such as text-based chats and image sharing).

In order for two peers to communicate it is necessary to have a signaling server. To share and view the necessary information to create a secure connection, signaling most often uses the Interactive Connectivity Establishment (ICE), Session Traversal Utilities for NAT (STUN), and Traversal Using Relays around NAT (TURN) protocols. The ICE protocol enables two browsers to connect and to create a secure connection using Session Description Protocol (SDP). ICE requires quick access to a peer's ICE candidates. A list of ICE candidates usually includes information like the IP address, port, and transport protocols that will be used in the secure WebRTC application. ICE can obtain this information with the help of the STUN and TURN protocols. A STUN server is located on the public internet and is capable of seeing the public IP address and port for your browser. Sometimes a STUN server fails to get around the Network Address Translation (NAT) firewall and is unable to obtain the public IP address. In these cases, TURN servers are used. TURN server job is to act as a relay server by receiving one peers media information, video, and audio, and promptly delivering it to another peers computer. This is different than the direct browser-to-browser exchange through a STUN server and is not a true peer-to-peer connection. ICE can now work and pass the information from the STUN server and incorporate it into the RTCPeerConnection API. At the completion of these steps, WebRTC will allow you to share video, audio, and text data across that channel for real-time communication [47].

3.3.2 Kurento

Kurento [48] is a WebRTC media server and a set of client APIs making simple the development of advanced video applications for browser and smartphone platforms. Kurento Media Server features include group communications, transcoding, recording, mixing, broadcasting and routing of audiovisual flows. As a differential feature, Kurento Media Server also provides advanced media processing capabilities involving computer vision, video indexing, AR and speech analysis. Kurento modular architecture makes simple the integration of third party media processing algorithms (such as speech recognition, sentiment analysis, face recognition, etc.), which can be transparently used by application developers as the rest of Kurento built-in features. Kurento vision is based upon openness, simplicity and standards [49].

3.3.3 OpenVidu

OpenVidu [50] is a platform to facilitate the addition of video calls in any web or mobile application. It provides a complete stack of technologies very easy to integrate in any application. OpenVidu wraps and hides all the low-level operations. The main goal it pursues is to provide a simple, effective, easy-to-use API so you can forget about WebRTC, ICE candidates and media server development. It is based on WebRTC and Kurento using many, but not all, of their features. It is an open source tool available for many platforms (Chrome, Firefox, Safari, Opera, Edge, Android, iOS and desktop apps) and is compatible with several frontend (JavaScript, TypeScript, Angular, React, Vue.js, Ionic, etc) and backend technologies(it exposes a REST API to allow any backend technology to be used) [50].

3.4 Chapter Conclusion

Looking at all the technologies that were analysed in this chapter it is possible to see that many of them could be used effectively for the purposes of this thesis. In the MR tools category the simpler tools like ARCore, ARToolKit and AR.js can be used in this project but each carry certain limitation that need to be carefully considered. While the bigger tools such as Wikitude and Vuforia have a lot more possibilities, their complexity can become a problem when trying to pair them with a videoconferencing tool. When it comes to the videoconferencing tools the foundations of the tools are always the WebRTC protocol because it is free and widely used, so the question becomes should we use tools that simplify the use of WebRTC such as OpenVidu, even though it does not allow for much freedom in the implementation process, or try to implement the WebRTC protocol directly while handling the necessary servers. The best way to answer all this questions is to use the research done in this chapter and to start experimenting with a few choice tools to see if they can work as intended and if they interact well with each other.

Chapter 4

Experimentation

After researching the various possible tools to use in this thesis, and deciding which seemed like the best choices, it was necessary to try out these tools to verify their functionality and to better understand how they can fulfill the requirements of this project. The MR tools chosen to be experimented with were Vuforia, for its many features and packages, and Ar.js for its simplicity, the fact that its web nature gives many advantages that were previously discussed and that it would pair well with the videoconferencing tools. The chosen videoconferencing tools were WebRTC tools that allowed us to use Vuforia in a video conference and OpenVidu because it facilitates the usage of WebRTC and would pair well with AR.js. This chapter starts by describing the experiments done with Vuforia and some of its tracking methods, as well as with AR.js. It also describes the choices and experimentation done with the researched videoconferencing tools that would best pair with the MR tools. Since the chosen tools only work with their respective pairs the chapter describes the experimentation with a MR tool, then with the corresponding videoconferencing tool and the integration between the two, and only then moves to the next MR tool.

4.1 Vuforia Experimentation

Vuforia has many features and tracking methods available for use. Since it is not clear which of this tracking methods would be best for our project, we implemented simple applications to gain experience with the development framework provided by Vuforia. At this juncture an attempt was made to experiment with all available tracking methods that Vuforia provides. Unfortunately some methods require certain tools and programs that were either paid or hard to acquire, so experimentation with them was not possible.

4.1.1 Image tracking

Image tracking allows for previously known images to be detected, tracked and augmented, this includes 2D images, cylinder targets (which are images wrapped onto objects that are cylindrical or close in shape like beverage bottles and soda cans) and several 2D images at once (multi-targets) which are arranged into any arbitrary arrangement of planar surfaces or into regular geometric shapes (e.g. cubes). After having a target image it needs to be uploaded into Vuforia's Target Manager (online tool that allows for the management and creation of databases and targets for Vuforia). Finally the image can be download from the database into the used application in the correct format to be used as a target.

Vuforia provides in its documentation the starter tools and guides in order to set up and try some of its features. Image tracking is easily deployed using Vuforia Engine and a sample. To experiment this feature I used Vuforia Engine for Android native development. Following the guide in Vuforia's documentation (https://library.vuforia.com/) it was necessary to download and install Android Studio IDE, Android SDK, Android NDK and Vuforia Engine for Android. After setting up this environment I downloaded Vuforia's sample for Android and built the sample using Android Studio and added a license key to the code (this license key was given when creating a free account in Vuforia). In order to run the sample on Android it was also necessary to enable developer settings such as installing apps from unknown sources and USB debugging. When running the sample on Android a menu opens up where you can select "Image Target". By selecting it and pointing the camera at a predefined image, a 3D model of an astronaut will appear (example model used in sample). Figures 4.1 and 4.2 show the sample running. In order to run the application using our own image targets, those assets would have to be uploaded into Target Manager and then download into the right place inside the sample application.

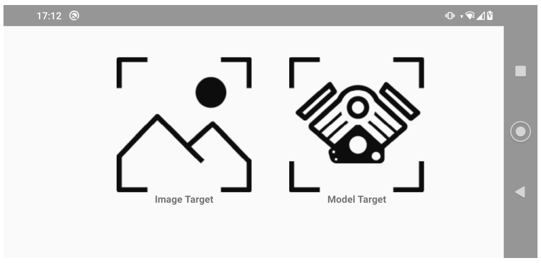


Figure 4.1: Sample application on native Android menu.

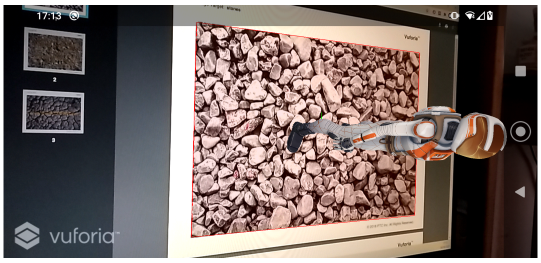


Figure 4.2: Sample application using Image tracking.

4.1.2 VuMarks

VuMarks are a specialised type of image target, they are a sort of bar code that is able to to encode data and to serve as a trackable AR target, example in figure 4.3. They are highly customizable and one design can be used to support multiple instances. Vumarks have very specific requirements in terms of design and in order to create one certain rules need to be followed. All this requirements are listed in Vuforia's documentation and they even have a plugin that can be used in Adobe Illustrator to help with designing.

Vuforia provides samples to experiment with VuMarks only in a Unity environment, so in order to experiment Unity was downloaded and installed. After this, all that was required was to download Vuforia's Core samples from the Unity Asset Store, to import Vuforia Engine SDK into Unity and it was necessary to add the license key into the project. The documentation regarding how VuMarks worked in Unity was then read in order to understand its implementation. Finally, the project was built for Android and executed. In the App a menu appears with several samples available, the VuMarks option can then be selected and the camera pointed at the example VuMark used in the sample. The example can be seen running in Figure 4.4. As can be seen in this Figure, a model of an astronaut is being added to the screen using MR together with a rectangle that is attached to the astronauts head with extra information. This rectangle is then added to the screen as can be seen in the bottom part of the Figure. This last part showcases how the VuMarks can pass along extra information to the users alongside the model unlike a normal image target.

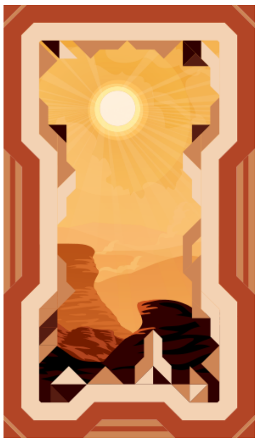
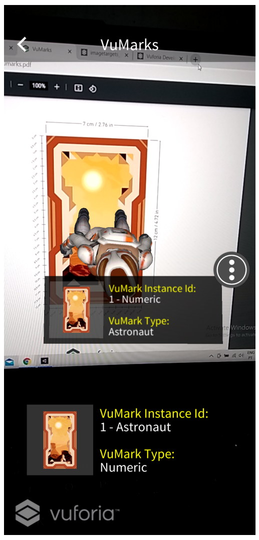


Figure 4.3: Example of a VuMark.



VuMark Instance Id:
1 - Numeric
VuMark Type:
Astronaut

VuMark Instance Id:
1 - Astronaut
VuMark Type:
Numeric

Figure 4.4: Sample application using VuMark.

At a later time several custom VuMarks were created and tested together with the default VuMarks to see their detection range. The results showed us that even with a custom VuMark, made to be easily detected, the detection range never surpassed 2.5 meters. This is not an ideal range to have, we would want somewhere over 3 to 3.5 meters so that when using this tool we do not have to get the camera close to the marker.

4.1.3 Object tracking

Object tracking uses object targets, which are created by scanning an object with the Vuforia Object Scanner (one of the tools available to support development), to track and add AR elements, this is the best option for detecting and tracking intricate 3D objects such as toys.

In order to experiment with object tracking there were two options. Either we printed the blueprints for an object that could be assembled and then used as a target, because the Unity samples already have its Object Data file. Or we could use the Vuforia Object Scanner to create an Object Data file from an object of our choosing. We chose the latter, so we downloaded the application and installed it in an Android phone. Firstly we had to print an Object Scanning Target image that is used by the application in order to know the scale the object that is being scanned has. We then placed the object in the Object Scanning Target image and used the application to start scanning (in this part, there are also several tips in the documentation about lighting and background that help with the scanning that we were not able to reproduce). The scanning process works by moving the camera around the object to capture as many angles as possible. After this it's possible to test the results within the application and finally to export the resulting Object Data file into Vuforias Database in their website.

With the Object Data file in Vuforias Database it was possible to load the target into the application and with a few changes in the code we managed to use the object as a target for tracking. This is shown in Figures 4.5 and 4.6.

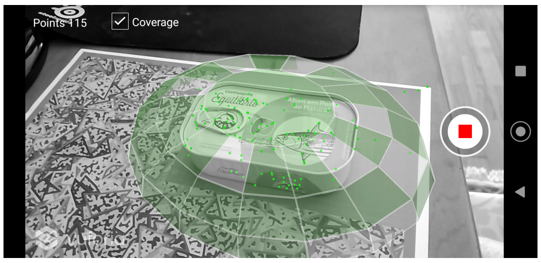


Figure 4.5: Using Vuforia Object Scanner to create an Object Data file.

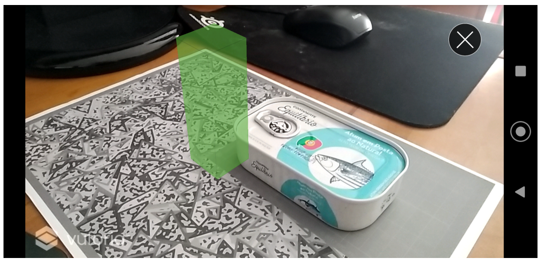


Figure 4.6: Sample application using an object as a target.

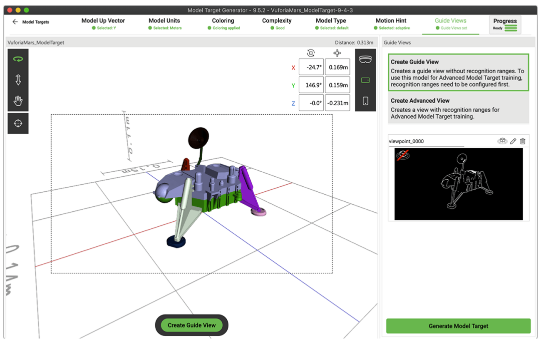
4.1.4 Non experimented tracking

Unfortunately there were several tracking options that were very interesting but could not be experimented with because they either required extra hardware or software that had to be paid. Nevertheless they were given a good in depth look so that it could be known what this features required and how they worked.

Model tracking

Model tracking uses model targets and it allows the engine to recognize objects by using pre-existing 3D models and by tracking its shapes. Model tracking is capable of tracking an object without having the whole object in view. Normally this tracking starts by holding the camera at a particular angle and distance to the object, this is called a Guide View (the app will draw in the screen an image showing an approximation of this information in order to help). This tracking is preferable for big objects like industrial equipment, vehicles and home appliances.

To use model tracking it is necessary to have access to a 3D model data of the object such as a 3D CAD model. Unfortunately we did not have access to an object an its 3D model data so it was impossible to try this type of tracking, nevertheless a description of the steps required to use model tracking will be made [51]. The first thing that needs to be done is to create a model target from the 3D model data, and to do this Vuforia has a supporting tool called "Model Target Generator" (MTG) [52], we can see an example in Figure 4.7. MTG can be used in Windows or MacOS to load in the 3D model data, afterwards certatin atributes need to be checked and verified, like the models orientation, its size, color and complexity. Then a few case uses choices need to be made like defining the model type and its motion hint (if its static or not). Finally a Guide View needs to be defined and the model can be generated and added to the target database, an example of the application using guide view to start tracking can be found in Figure 4.8. Vuforia also provides another tool called "Model Target Test App" that allows you to test your model and verify how trustworthy and accurate it is or you can import it into Vuforia's sample and try it out.


Model Target Generator - 9.5.2 - VuforiaMars_ModelTarget-9-4-3

Model Up Vector: Selected: Y
Model Units: Selected: Meters
Coloring: Coloring applied
Complexity: Good
Model Type: Selected: default
Motion Hint: Selected: adaptive
Guide Views: Guide Views set
Progress: Ready

VuforiaMars_ModelTarget — Distance: 0.313m

X -24.7° 0.169m
Y 146.9° 0.159m
Z -0.0° -0.231m

Guide Views

Create Guide View
Creates a guide view without recognition ranges. To use this model for Advanced Model Target training, recognition ranges need to be configured first.

Create Advanced View
Creates a view with recognition ranges for Advanced Model Target training.

viewpoint_0000

Figure 4.7: Using Model Target Generator to create a model target. [52]

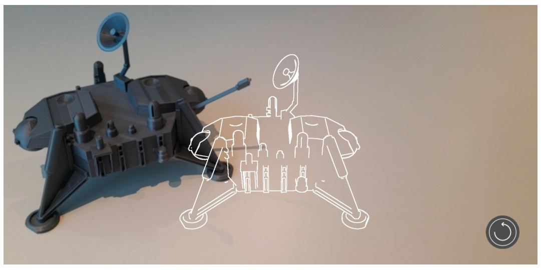


Figure 4.8: Example of an application using a model as a target, where the guide view is shown as a white outline. [51]

Environment tracking

Environment tracking is done by using a 3D scan as an accurate model of a limited area and then allowing the user to track and augment these areas and spaces. This enables creating games and navigation applications that are all using the surroundings as interactive elements. Offices, factory floors, apartments, public spaces, museums, and many more areas are ideal sites for Environment tracking.

In order to create area targets a digital model needs to be obtained from one of the 3D scanning technologies supported by Vuforia. These technologies are ARKit enabled devices with inbuilt LiDAR sensors, Matterport™ Pro2 3D camera, NavVis M6 and VLX scanners, and Leica BLK360 and RTC360 scanners. Vuforia provides the "Vuforia Area Target Creator App" [53] that uses ARKIT with LiDAR sensors to create a area target and since it uses ARKIT it is only available for iOS, example in Figure 4.9. The other options use specific scanners and cameras that are quite expensive. In Vuforia documentation [54] you can find recommended ways and tips on how to use any of these scanners. The only one that could possibly be used to experiment with environment tracking would be with "Vuforia Area Target Creator App" but we did not have access to an iOS device. Nevertheless a descriptions of the steps required to use Environment tracking will be made. With the area target created, all that is left is to use another Vuforia tool called "Area Target Generator" to generate the dataset required to use with an application. This process is straight forward, you upload the area target and the generator outputs the files required. Finally it imports the dataset into your project and with a few code additions to the sample it is possible to experiment with the environment tracking or use the "Area Target Test App" from Vuforia to give an idea of how good the scan was, an example of it running can be seen in Figure 4.10.

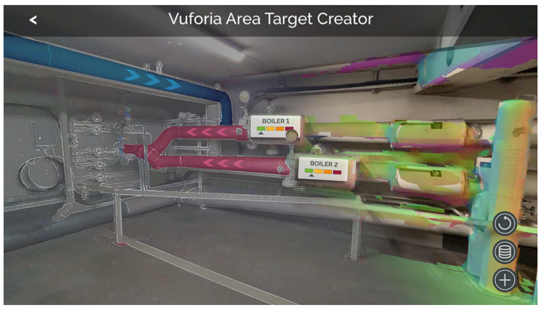


Figure 4.9: Using Vuforia Area Target Creator App to create an environment target. [53]

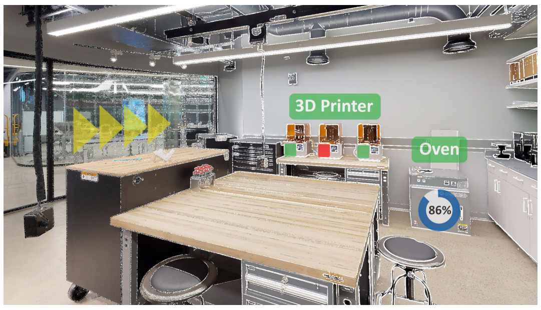


Figure 4.10: Example of an application using the environment as a target. [54]

Instant tracking

Instant tracking is where the engine can recognize horizontal surfaces, such as floors and tabletops, and then add digital content on top, it also enables you to place content in mid-air using Anchor Points [55].

Vuforia provides instant tracking samples only in Unity environment, so in order to experiment it was downloaded and installed in Unity. After this, all that was required was to download Vuforia's Core samples from the Unity Asset Store, to import Vuforia Engine SDK into Unity and it was necessary to add the license key into the project, an example can be found in Figure 4.11. The documentation regarding how instant tracking worked in Unity was then read to understand its implementation. Finally, the project was built for Android and was executed. In the App a menu appears with several samples available, we can then select the Ground Plane option but unfortunately the android phone used was not compatible with this type of tracking, as it did not meet the requirements. An example of this type of tracking being used can be seen in Figure 4.12.

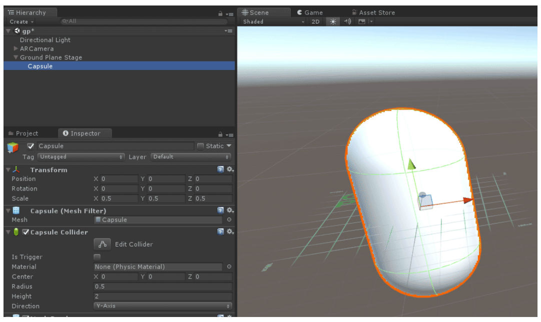


Figure 4.11: Using Unity to create an object to be placed in the ground. [55]

Figure 4.12: Example of an application using the ground as a target. [55]

4.2 WebRTC Experimentation

After experimenting with Vuforia Engine and having used Unity as a development tool it was clear that Unity was a very interesting option to pursue. Unity uses a graphical interface to help development which is an advantage when using MR elements, also the fact that it allows development for most platforms (Windows, iOS, Android, etc) with minimal coding chance is also a plus. This interest in using Unity made us research into options to use WebRTC in Unity. Two third party packages for Unity that implement WebRTC were found. The first one called "MixedReality-WebRTC" is a collection of components that help MR app developers to integrate peer-to-peer audio, video, and data real-time communication into their application and improve their collaborative experience. The second one is called "WebRTC" and is a much more general all propose use of WebRTC in Unity making it a bit more complex than the first one. Another package worth mentioning is the "Unity Render Streaming" package that is based on the previous "WebRTC" package and allows for the streaming of rendered video and audio between machines, it also allows for remote control. In this project the packages "MixedReality-WebRTC" and "Unity Render Streaming" were used in the experimentations.

4.2.1 MixedReality-WebRTC

In order to download and install the package into Unity it was necessary to use "Mixed Reality Feature Tool", a free Microsoft utility to manage MR packages for Unity [56]. After installing and verifying the package the first thing to do was create a peer connection using the given library. Then since we are using WebRTC an external server for signalling was necessary, we used a signaller called "NodeDssSignaler" which is very simple because it has no security or authentication (for this reason there was no need for STUN/TURN servers). It was also necessary to create a connection in Unity to this signalling server. Then a local video and audio objects had to be created and linked into a video player that we also had to create. After that the remote video, audio and video player also had to be created and connected. Finally the "NodeDssSignaler" had to be configured inside unity with the local and remote peers and finally it was possible to run two instances of the application, with one having the option to initiate the connection, and have a working WebRTC connection. Figure 4.13 show an example of a simple exchange of camera feeds in between two user using this package.

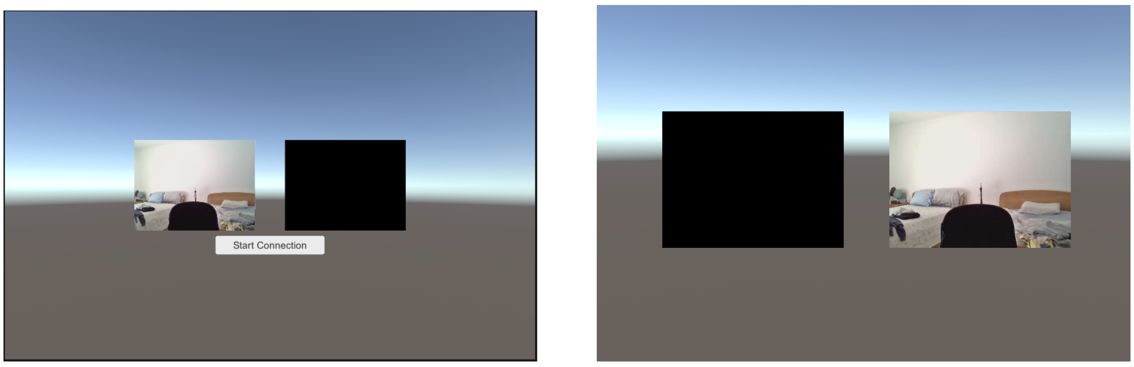


Figure 4.13: One user is creating a WebRTC connection (left image) and another user (right image) is connecting to an existing WebRTC connection (the panel with a web cam image is the local camera feed for the user creating the connection, right image, while the black panel is the local camera feed for the user that is connecting, left image).

4.2.2 Unity Render Streaming

To use this package we must first download and install it using Unity's package manager which is very easy and standard [57]. In order to set up a sample scene for testing we have to add several components to the camera object in unity. This components are "Render Streaming" that allows us to change the basic settings of the connection such as choosing the signaling type and URL, "Broadcast" that delivers the stream to multiple peers and "Camera Stream Sender" that refers to the camera feed and delivers it to the "Broadcast" to deliver to peers. To be able to run this sample we must also use a signaling server that the developers of this package also provide, running this server allows us to use a web page on localhost to view the feed from the app on unity. Figures 4.14 and 4.15 show an example of the camera feed being captured in a Unity application and sent to a web browser.

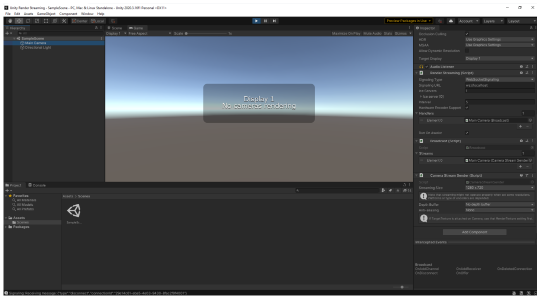


Figure 4.14: Camera feed being sent from the unity app to the server.

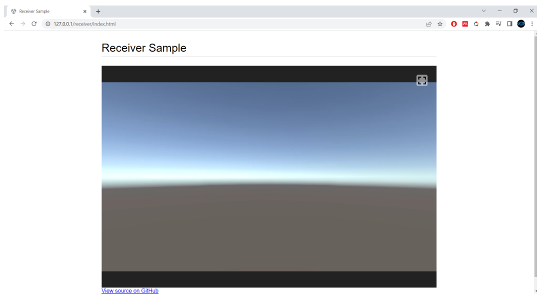


Figure 4.15: Camera feed from the unity app being displayed on the web browser.

4.3 Integration of Vuforia with Render Streaming

After testing this tools separately it was important to understand if they could be made to work together to achieve what we set out to do. So, using Unity we installed the Vuforia engine and the "Unity Render Streaming" package to try using both at the same time. We choose the package "Unity Render Streaming" as after testing this one managed to be compatible with the way Vuforia is implemented in Unity.

The initial installation of this packages followed the same steps as described before with Vuforia being installed using Vumarks as the tracking method. Afterwards the components from Vuforia and from "Render Streaming" were both placed on the same camera object and a few tests were done. This tests confirmed that the two components can coexist in the same object proving that the integration of this tools is possible. Of course much more work would have to be done to have this tools working perfectly together, one immediate apparent problem is the fact that the resolutions of the camera that is sent to the peers does not match the resolution of the video where the 3D objects created by Vuforia appear, but this initial results are sufficient for us to move into making a decision about what tools to use in creating our project.

4.4 AR.js Experimentation

AR.js as mentioned before has three different types of tracking options that can be used. The location based tracking in AR.js is used mostly outdoors, our application is intended to be used inside a building so it would not make sense to experiment with this type of tracking. The other two types of tracking are Image tracking and marker tracking and while this two have differences in how they work, in practice they are similar (they both use a 2D printed image as a marker) so only the marker tracking was experimented with as this one is usually a more reliable and controllable tracking method.

Running a simple sample of Ar.js is quite simple [33], all that is required is to create a web project with an HTML page on it. Afterwards you need to add one line of code to import the render library used by AR.js (this can be A-Frame or three.js, in this experiment we used A-Frame). Next a small snipped of code provided in the documentation of AR.js needs to be added to the HTML page. Finally all you must do is run the web server and point the camera at a sample marker provided in the documentation. An example can be seen in figure 4.16.

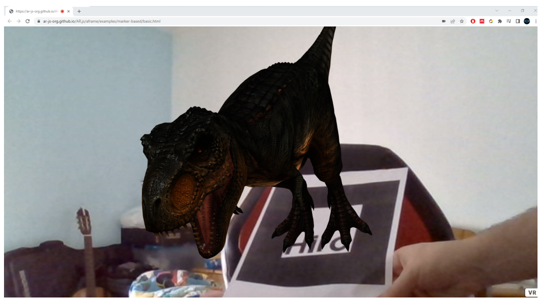


Figure 4.16: Web page open with AR.js and a marker being tracked while adding a 3D model on top.

4.5 OpenVidu Experimentation

In order to run an OpenVidu server [50] it was necessary to first prepare a Linux environment. To do this, the software VirtualBox was used where we created a virtual environment with a Linux operating system (the chosen system was CentOS). Three different libraries had to be installed to run the OpenVidu servers, we had to install git in order to download the samples (both the WebRTC server and the sample web page), we also had to install the library "http-server" to be able to run the web page and finally we had to install the docker engine to run the OpenVidu server that contained the WebRTC server.

Afterwards all that needed to be done was to run the docker with the server following the instructions in the documentation and run the web server with the library "http-server". To verify that the sample was functioning properly two web browsers were opened connecting to "localhost" with the port 8080. With the sample running we can see the connection being made between the two web browsers. An example can be seen in figure 4.17.

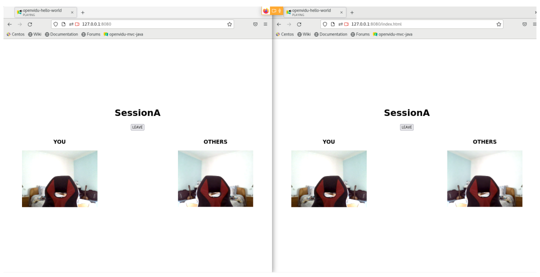


Figure 4.17: Camera feed being sent from one browser to another using OpenVidu.

4.6 Integration of AR.js with OpenVidu

Integrating this two tools together was actually quite simple thanks to the simplicity that comes from the MR tool. All that AR.js needs in order to function is the addition of the library and a few lines of cone in the http file of the web page. So this additions were made to the sample used before in the OpenVidu experimentation.

After doing some testing it was possible to see that the web page was able to send video to the peer while at the some time track 2D images to add 3D object, proving that this tools can work together. Unfortunately some modifications still need to be made to the code for it to be viable as one of the first problems that can be seen is that the video that gets transmitted to the peers does not show the 3D object. But for this section of the thesis it proves that this can be a possible solution to the project.

4.7 Chapter Conclusion

Having used and worked with this tools we have gained a better understating about how they function, how they can be modified and their limits. With the experience and knowledge provided by this experimentation it is now possible to make a well informed decision about how much is possible to do with them and what tools will best fit with our use case, of course first we must define it.

Chapter 5

Requirements Analysis

This chapter defines the use case and the requirements for this project. This description will start with the use case to ensure the reader understands the decisions that are made moving forward. Next, the requirements of the applications are described and organized. Followed by how the project will function with some mockups, that were developed during the implementation, to better visualize the intended application.

5.1 Use Case

The proposal of this project specifies only that a prototype of a generic platform to support MR within a video conference must be created. In order to create a more guided and complete prototype a specific use case was defined.

The idea is to create a platform so that a company may use it in order to make a live visit to one of their building (warehouse/factory/headquarters), while using the MR elements of the application to make a more engaging and informative presentation. The presenter will be able to share the live feed of their device to the other participants in the visit. When the presenter wishes to show more details and information about a certain object (for example a machine in a factory) all that needs to be done is to point the camera at the machine (that will have some from of tracking in it) and an element of MR will appear and add that information to the visitors screen in an interactive way.

This work proposal has the support of the GLN group, a company engaged in the production both of technical molds of high precision and performance, and in highly technical level plastic injection. Our application will be geared towards the visit of a GLN factory while showcasing (using MR) their molds and plastic injection machines.

5.2 Requirement

To better define our use case it is necessary to present all of the conditions it must fulfill. This will give us a good idea of what the application needs to have and all the features that will be implemented. To do this the MoSCoW method has been used and can be seen in Table 5.1. Some of this requirements pertain to the applications videoconferencing tool such as the ability to mute/unmute and the chat functionality. Others are about the MR tool such as the recognition of the markers. A few are about the functionalities of a web application like the possibility of logging in with a link. And a good number of them are about the quality of the application like when we consider the performance or the accessibility of the application.

Must have	-Being able to share video and audio with other users -Good quality for video and audio -Integration of Mixed Reality into the Video conference -Able to track markers -Show Mixed Reality elements on top of marker -Have a left column where references of the markers will be added -After recognizing a marker a reference is added to the left column -Clicking a reference in the left column will expand the information -Presenter and visitor side to the application -Have good performance -Being accessible
Should have	-Be able to mute/unmute audio and video -Be able to login through a link -Be able to leave a session -Working in several browsers and platforms -Show Video conference participants -Have an indicator when a user is muted -Each user has an identifying tag -Presenter has a unique tag so users know easily who they are -Be able to change the video feed of the main window
Could have	-Have a chat for the Video conference participants -Have several possible session running at the same time -Presenter can control the extra information shown to the users.

Table 5.1: Table of requirements using MoSCoW method.

5.3 Description of the application

This application will have two versions, one for the presenter, which will have more control over the presentation, and one for the visitors, which will have some interactivity. The presenter side will be developed with mobile platforms such as smartphones and tablets in mind (since they are going to walk around the building with a device that needs to have a camera) while the visitor side will be developed with computers in mind such as desktops or laptops (since the visitor will most likely be sitting in a desk watching the presentation).

The way the MR elements are planned to be added to this project is by adding several markers that can be tracked by the camera from the presenter side of the application. The markers that will be added are preset images that are easy to track, this markers can at any point be replaced or changed inside the application. For each different marker that will be added there will be a 3D model associated with it, this model can be of anything that is necessary as long as it is in a predetermined format (this format will be defined during the implementation). For this project the idea is for the model to be a simple label that appears on top of the marker with the name of whatever we are trying to showcase (in our use case probably a type of heavy duty machine). In order to add interactivity there will be a column at the left side of the UI in which buttons will appear every time a new marker is detected. The idea is for each button to be unique depending on the marker found, and by pressing this button, in depth information will be shown on the subject to which that marker pertained to. This column, button and extra information is going to be implemented in both sides of the application (presenter and visitor) but the presenter will have some control over what is shown on the visitors side. This functionality can be visualised in the first mockups created for this application (Figures 5.1 and 5.2).

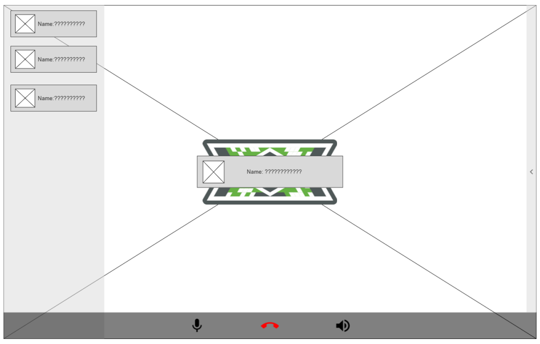


Figure 5.1: Mockup of the main view in the visitor.

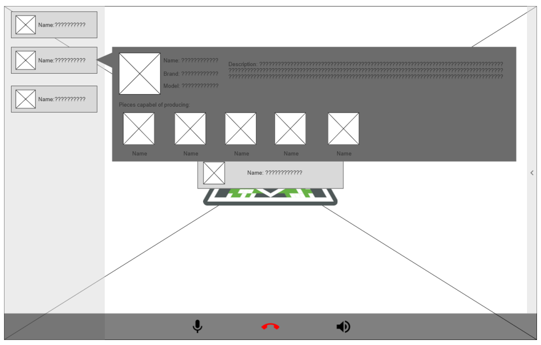


Figure 5.2: Mockup of the main view in the visitor with extended information.

The background image in the mockups (represented by a white square with two diagonal lines crossing at the center) is the live feed of the camera, where the presenter will point the camera at a marker (an example marker is shown in the mockups in the center [58]). While looking at this marker MR elements will pop up, this is represented by the text box in the center on top of the marker (Figure 5.1). In the leftmost side of the screen there is a semi transparent column where every time a marker is detected a small text box with the corresponding name will be added to the column. By clicking on this button in the column it shows more detailed information on the machine (Figure 5.2).

On top of all this the application will have all the normal videoconferencing functionalities such as muting, turning off video and chatting through text. Most of this functionalities can be seen as buttons in the more recent mockups (Figure

5.3) where the visual aspect of the application was cleaned and improved. It is also worth mentioning that a template for the panels where the extra information shows up was created (Figure 5.4).

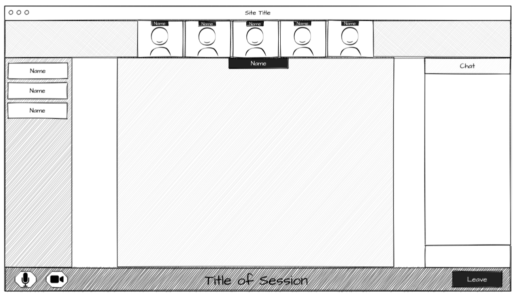


Figure 5.3: Mockup of the updated UI

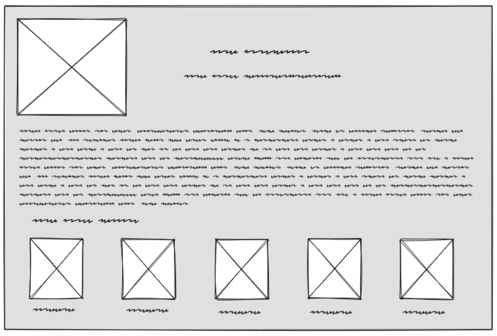


Figure 5.4: Extra information popup template.

All mockups that were created during the research and implementation of this project can be found in Appendix A. Note that there were three different points in time where mockups were created to represent the new ideas and improvements that were being added to the project.

Chapter 6

Architecture & Implementation

In this chapter of the thesis we start by detailing all the choices made based on the research done and the use case. Then the architecture of the project that was developed will be described alongside all the properties of the elements in it. Afterwards the inner workings of the application that was built will be showed in detail. This will be followed by all the functionalities that were added to the application while describing how they were added. Next a description and several images showcasing the application in it's complete form are shown. Finally the deployment of this application will also be described.

6.1 Choices for tracking/tools/platforms

After having a good grasp of what the use case for this project is and with all the research and experimentation done on the existing tools we now need to make the important decisions of what tools we are going to use for development, what kind of tracking will be used, for what platforms we will make the application, what kind of functionalities will it have, among other decisions.

One important decision in our use case will be the type of tracking that will be used. All the types of tracking studied in the experimentation are quite interesting and have their upsides and downsides. Image tracking is the best one that can be used in our project, for our use case we want the detection to immediate and to have a good detection range. Other solutions did not allow for this. Also having a printed target somewhere in the path of the presentation in order to satisfy this type of tracking does not seem to be a problem.

The next decision to make is what tools should be used in implementing this project. For the MR tool we experimented with two good possibilities, AR.js and Vuforia.

With Vuforia we can create an application that allows for a lot of different development environments with a lot of different possible functionalities. A problem with Vuforia when using Image tracking is that it does not have a large detection range (it ends somewhere around to 2 meter from the target), and this poses a problem in our use case. In a building with, for example, several big machines we want to present, we do not wish for the presenter to have to always come very close to the machines for the application to detect the target, we also do not know if the best place to put a target will always be accessible at close range to the presenter.

In the scenario with Vuforia we would use a Unity package to add the videoconferencing capabilities to the application. We have already, in the experimentation portion, found a usable package that can be integrated with Vuforia in Unity. While this package adds WebRTC properties to the application it does not give any solution to the server side of the videoconferencing (the creation of the signaling server and STUN/TURN server). This means that the servers would have to be defined and created manually.

As an alternative we can use AR.js to create the application. This tool is much simpler than Vuforia, with a few lines of code we can start using it without much complication. This means that the development will be much more streamlined and efficient. With AR.js we can also achieve a detection range in Image tracking of about 4 meters. This tools forces us to create a web application which has some advantages and disadvantages. Using AR.js also means using OpenVidu as a tool for videoconferencing. While this tool also uses WebRTC it already handles a lot of the hassle that comes with it. OpenVidu provides tutorials and samples on how to create all the necessary server for video conferences with security already in place. It also provide a few sample web pages that use OpenVidu that can be used as a starting point for creating the web application.

With all this considerations it was decided that the best fit for this project was the usage of AR.js and OpenVidu. This is because most of the value that Vuforia brought to the table was put aside for many different limitations associated with the project. So at the end AR.js and Vuforia could both do the job that the project required, but the fact that Vuforia by its nature is more complicated and does not have a large detection range, makes AR.js come out on top. What makes this a more obvious choice is that with AR.js we can use OpenVidu which solves many of the issues that arrive from creating a videoconferencing application.

Given the nature of web applications an effort will be made to make the application accessible to as many operating systems and browsers as possible. The development will have in consideration the variety of platforms that can be used to access the application such as PCs, phones and tablets.

6.2 Architecture

This project's architecture consists of a front-end where the browser client is found and a back-end where we have the Web App server, the OpenVidu server and the Kurento Media server. The browser client is where most of the work for this application went into, this is where we code and design most of the application, it is also from here that we can manage most of the properties from the video-calls and all the aspects of the MR elements. The Web App server is where we manage some of the most common yet necessary security aspects of an application, this includes being able to login with an account and managing all necessary tokens for a session. The OpenVidu server is a culmination of all the servers that a WebRTC application needs to function, so this server does the job of a signalling server and of a TURN/STUN server. Finally the Kurento Media server handles all the low level media flow transmissions and all the changes and different operations that can be done to it.

Both the OpenVidu server and Kurento Media server used in this project were provided by OpenVidu and the only changes that had to be made was the alteration of certain properties such as IPs and Ports. The servers communicate with each other through the use of WebSockets with the exception that the browser client uses WebRTC to communicate with the Kurento Media server. The architecture described here can be visualized in a diagram in figure 6.1.

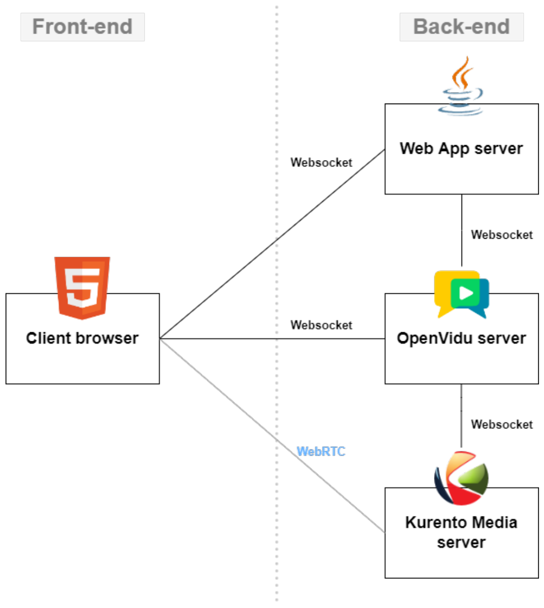


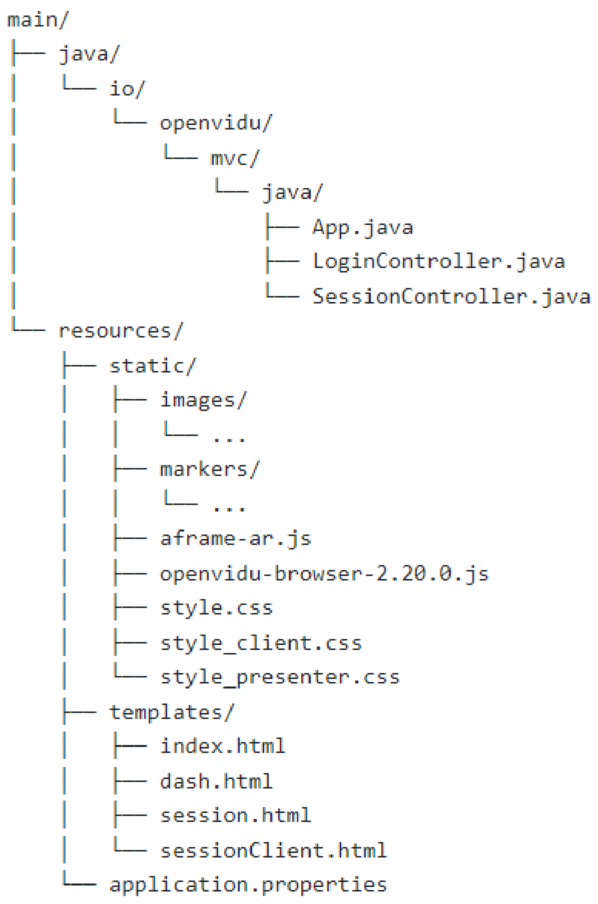
Figure 6.1: Architecture of the application.

The languages used to make this projects front-end were HTML, css and javascript, as is expected for a browser client. The front-end follows a traditional MVC (Model-View-Controller) architectural pattern for better logical organization. In the back-end, the Web App server was written in java because OpenVidu provided a template that explained the connections between the Web App server and the OpenVidu server and it was written in java.

It is also import to explain the files that all this components generate and how they are organized. The OpenVidu server and the Kurento Media server come in a docker-compose container (docker-compose is when you join two or more docker files to be run as one container) which makes them easy to identify and use, in the deployed final version their files are separate from the rest. The browser client and the Web App server generate many files that need to be explained. Inside the server running this application there is a folder that houses two more folders, one that has the files for the browser client (with the path "/main/resources") and one for the Web App server (with the path "/main/java"). Inside the browser client folder there is a file called "application.properties" that has all the variables that the application needs to communicate with the OpenVidu server, this includes the port, url, secret password and all ssl certificate properties. Alongside this file there are two more folders, one called "templates", that contains the HTML files, and another called "static" that has all the other files that are necessary.

In order to explain the HTML files that exist in the "templates" folder it is first necessary to explain the flow that our users are expected to experience when using this application. The users will first encounter a page where they will be able to use credentials in order to login into the application, afterwards they will encounter another page where they can decide what their nickname in the video conference will be and which session they wish to join and then they will finally see the page with the videoconferencing and MR elements. Knowing this we now know that we need a HTML file for the login page, which was named "index", a HTML file for the choice of session, which was named "dash" and two HTML files for the videoconferencing page, one for the visitors named "sessionClient", and one for the presenter named "session". In the "static" folder there are the used libraries, one for AR.js called "aframe-ar" and one for OpenVidu called "OpenVidu-browser-2.20.0". There are also the css files which contain the description of how HTML elements are displayed on the screen, the file called "style" include the css used for both the "index" and "dash" HTML pages, while the file called "style_client" has the css for the visitor side and "style_presenter" has the css for the presenter. Finally, still in this folder there are two more folders, one with all the images used in the application and another that has all the 3D models used in the MR part of the application, this folder being called "markers".

The Web App server generates three different java files, the first called "App" that initializes the server, the second called "LoginController" that handles the login portion of the application and the final one is called "SessionController" and is the file that handles all the operations that happen during a session of a running video conference. This organization can be better visualized in a diagram in figure 6.2.



```
main/
├── java/
│   └── io/
│       └── openvidu/
│           └── mvc/
│               └── java/
│                   ├── App.java
│                   ├── LoginController.java
│                   └── SessionController.java
└── resources/
    ├── static/
    │   ├── images/
    │   │   └── ...
    │   ├── markers/
    │   │   └── ...
    │   ├── aframe-ar.js
    │   ├── openvidu-browser-2.20.0.js
    │   ├── style.css
    │   ├── style_client.css
    │   └── style_presenter.css
    ├── templates/
    │   ├── index.html
    │   ├── dash.html
    │   ├── session.html
    │   └── sessionClient.html
    └── application.properties
```

Figure 6.2: File path of the application.

6.3 Session Management Implementation

As said previously, the Web App server is comprised of three different java files "App", "LoginController" and "SessionController". This server is built using the Java Spring Boot tool which makes developing web application with Spring Framework faster and easier. The "App" file is very small and simple as all it does is initialize the server and function as an entry point for the application.

The "LoginController" file handles the login and logout operations. This controller is used when the server receives a POST operation with the username and password when a user enters the application (in the "index.html" page) and wants to login (the server is receiving this information from the front-end). What this controller does first is check if the user is already logged in, if so it redirects the user to the next html page ("dash.html"). If not and the user is logging in for the first time the controller checks that all the necessary parameters are correct and sets an HttpSession for the new user (HttpSession is a variable that holds all the necessary information so that if they are already logged in, there is no need to validate again) and it then sends the user to the next html page ("dash.html"). If at any point the user provides incorrect information the user is sent back to the "index.html" page. They are also sent there if they logout of the application. Figure 6.3 summarizes the logic that the controller uses.

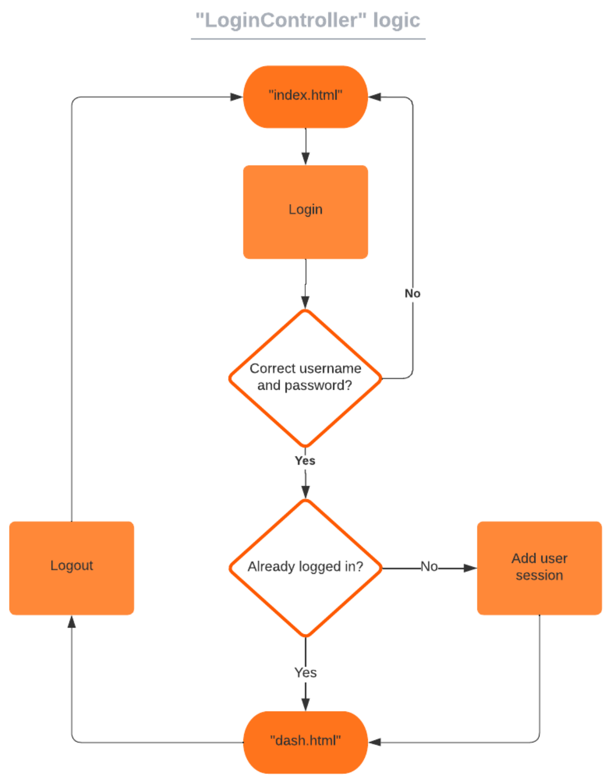


Figure 6.3: Diagram with "LoginController" logic.

Lastly the "SessionController" file handles the tokens used by OpenVidu to create and maintain sessions, it also stores active video conferences and the users connected to them. When a user is in the "dash.html" page they then want to either start or join a session, and that is where this file comes in. When the user gives the requested session name and nickname, the front-end sends a POST operation with this information. The controller then connects to the OpenVidu server, creates a user and checks if there is already a session created with given name. If there is not, then the controller creates a new OpenVidu session and sends the required tokens and the user to that videoconferencing session, if there is then the controller simply adds the user to that session by sending all the information required. This controller handles the connection when a user is exiting a session by closing it. It is also this controller that makes the distinguish between visitor and presenter, if the user is a visitor at the end it goes to "sessionClient.html", if it is the presenter it goes to "session.html". Figure 6.4 summarizes the logic that the controller uses.

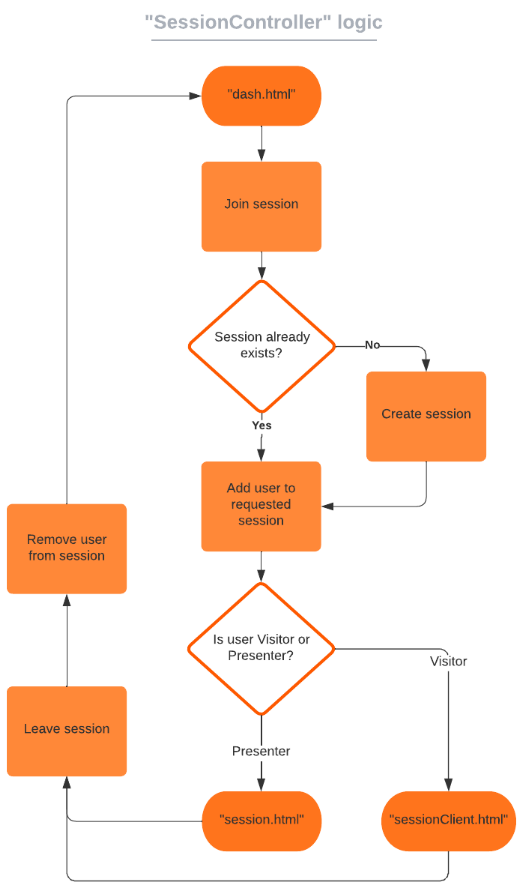


Figure 6.4: Diagram with "SessionController" logic.

6.4 MR Elements Implementation

The AR.js portion of the code is exclusively found in the "session.html" file as only the presenter side of the video conference will be able to detect markers and show 3D models on top of them. The idea is to stream the video from the presenter with the MR elements already on it to the OpenVidu and Kurento Media server to be sent to the visitors. In order to do this we need to add the required libraries to the html file.

Since we are using "A-Frame" as a method within AR.js to display the MR content we need to create a scene inside the html file using the tag "<a-scene>". It is in this tag that we must first set several necessary proprieties such as saying that this scene will contain AR.js elements, that the video should be fetched from the webcam, and other less important variables. Inside the scene tag we also have to have a camera element, we create it using the tag "<a-entity>". The last elements we had to add to the scene were the chosen markers that the application needs to recognize, for this we use the tag "<a-marker>". In each of this markers we need to say what kind of markers they are (in our case we used barcodes as a simple example), what the marker looks like (in the case of barcodes what they look like is defined by a number ranging from 0 to 32), and what are the 3D model that should be displayed. To tell the application what the 3D model looks like we had to create an "<a-entity>" tag and in it say the position and scale the model should have in relation to the marker and what outside file contains the model, in this case we used GLTF (GL Transmission Format) models as was recommended by the documentation of AR.js. In the following figures (Figure 6.5 and Figure 6.6) we can see what an example marker and model will look like.

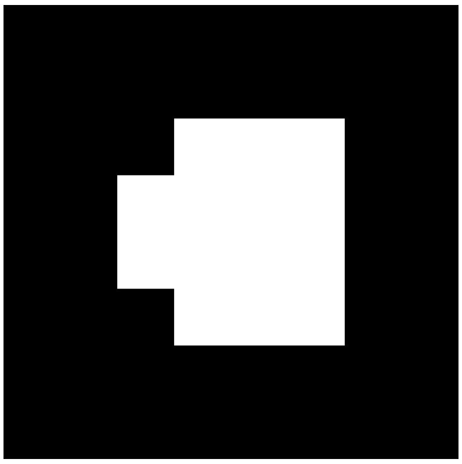


Figure 6.5: Example of a barcode marker.

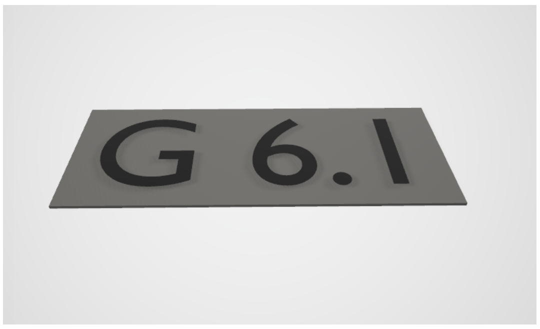


Figure 6.6: Example of a model used in our project (a simple tag with a letter and number).

One final change that had to be added to the code relating to AR.js was that in order for the application to be able to recognize each markers appearance in the video feed individually and in order to create events relating to this markers it was necessary to create components for each one. This was done using javascript where for each marker a component was registered using different names and this names were added in the "<a-marker>" tags to connect the two. Now with this components created we can add events to the application regarding individual markers showing up on the video feed.

6.5 OpenVidu Implementation

The OpenVidu implementation refers to the session creation and management part of the application as it is each session that provides videoconferencing. We have said previously that it is in the Web App server that the sessions get created, which is true but they call on the front-end to actually run the session, so part of creating a session happens in the "session.html" file, more specifically in the javascript portion of this file.

This file starts by initializing a session using all the info that comes from the back-end. Then it defines all the events that the session is supposed to handle, some of the default events are when an exception is found or when the stream from a user is destroyed. One of the most used events in the creation of this application was the usage of the "signal" event. This event allows for some code to run while notifying some or all of the peers in that session that this event is running, this allowed for many of the interactions necessary for the MR elements of the application.

It then proceeds to create a connection to the media server to start streaming video to its peers, this connection is either used to start a new session or join an existing one, regardless all users using this application go through all these steps. It is in this section that we can change many variables of the session, such as the video and audio input, the resolution of the video, what information is shown, etc. Alongside all this, the javascript also contain several functions that help run and support the creation and management of the sessions. Figure 6.7 summarizes this information.

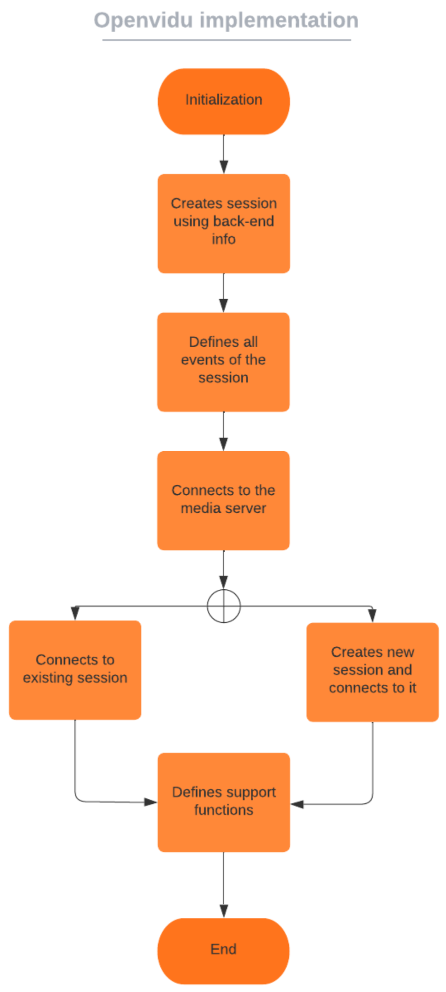

Openvidu implementation
Initialization
Creates session using back-end info
Defines all events of the session
Connects to the media server
Connects to existing session
Creates new session and connects to it
Defines support functions
End

Figure 6.7: Diagram with OpenVidu's implementation logic.

While adjusting this portion of the application we ran into a significant problem with our solution. When we tried to send the video feed from AR.js to the peers using the options available while creating the connection to the media server we realized that the 3D elements were not showing up in the visitor side of the session. What we discovered is that AR.js does not add the 3D elements on top of the video feed, it adds it to the canvas element of the html page. To solve this problem we decided it would be simpler to add the video feed to the canvas, so that we could have the feed and the elements in the same place, and then send a feed of the canvas to the peers. This was done using a function that added the video to canvas and at the same time made the video play in the canvas, so that we would not end up with a frozen image. With this all that was left was to add the video track of the canvas as the video input of the connection of the session.

6.6 Interactivity

The most important functionality that needs to be explained is the one that adds interactivity with the use of MR. As explained in a previous chapter the MR elements are added using AR.js and when they are detected buttons are created. The buttons can then be pressed to show extra information regarding the particular marker that was detected. In figure 6.8 we can see an example where the area in question is indicated with a red outline and the buttons with a red arrow. How this functions in the application is that, in the components we created for each different marker we add an event listener that detects when that specific marker is seen by the camera. Inside this event listener we then add an OpenVidu signal that tells every single participant in that session that a marker was found and says which one. With every participant knowing what marker was found they then create a visible button in the left column and an invisible panel that has all the extra information. This button can be clicked by any visitor and the panel will become visible in front of the main video.

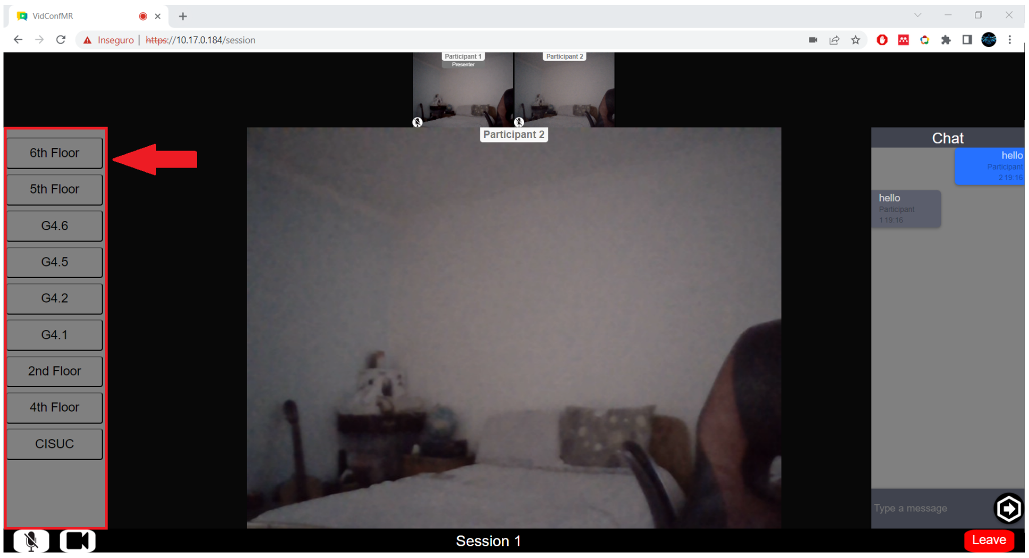
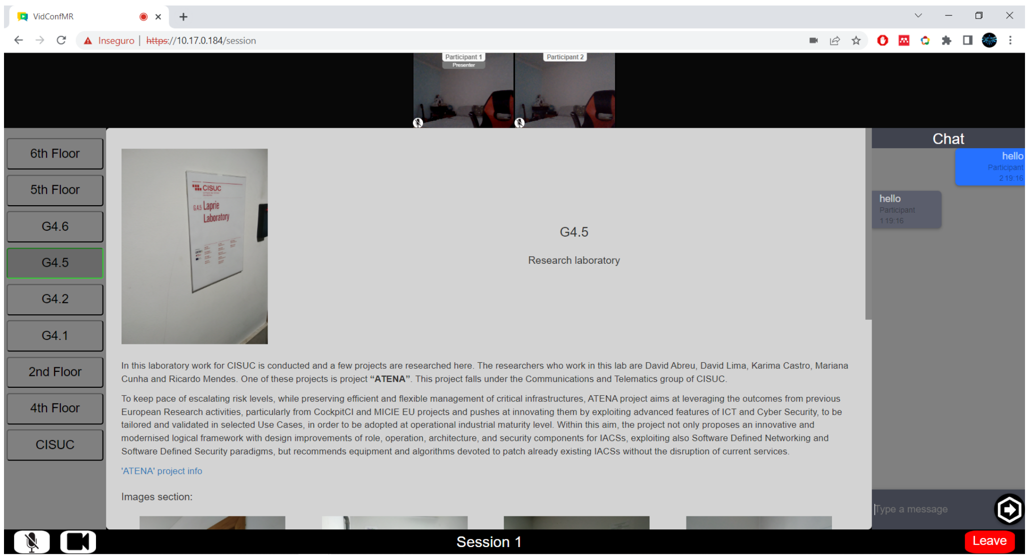


Figure 6.8: Example with red outline and arrow pointing at left column and buttons created with the use of MR.

This panel only opens to the visitors that click the button. Visitors have the autonomy to look at and read the information whenever they please. For the presenter this is a bit different. A small code was created to make it so that whenever the presenter clicks one of the buttons a signal is sent to every visitor, if the visitor receiving the signal has no other open panel then the panel the presenter opened also opens to the visitor, if not then it does not open. When the presenter closes a panel then if the visitor is in the same panel as the presenter it closes, if not then nothing happens. Outside of this conditions when the presenter opens one of the panels the button corresponding to that panel will have an outline in the visitors side, this makes it so that the visitors are always aware of what panel the presenter has open, when the presenter closes the panel the outline goes away. In figure 6.9 we can see an example where the fourth button in the column is pressed by the presenter and it's info visible becomes visible to all visitors. It was taken into consideration the fact that the presenter will most likely pass over the same marker several times, because of this we made it so that no button can appear twice in the column. We also made it so the newest buttons are always on top and that when the column is full, the older buttons start being deleted.



G4.5

Research laboratory

In this laboratory work for CISUC is conducted and a few projects are researched here. The researchers who work in this lab are David Abreu, David Lima, Karima Castro, Mariana Cunha and Ricardo Mendes. One of these projects is project **"ATENA"**. This project falls under the Communications and Telematics group of CISUC.

To keep pace of escalating risk levels, while preserving efficient and flexible management of critical infrastructures, ATENA project aims at leveraging the outcomes from previous European Research activities, particularly from CockpitCI and MICIE EU projects and pushes at innovating them by exploiting advanced features of ICT and Cyber Security, to be tailored and validated in selected Use Cases, in order to be adopted at operational industrial maturity level. Within this aim, the project not only proposes an innovative and modernised logical framework with design improvements of role, operation, architecture, and security components for IACSs, exploiting also Software Defined Networking and Software Defined Security paradigms, but recommends equipment and algorithms devoted to patch already existing IACSs without the disruption of current services.

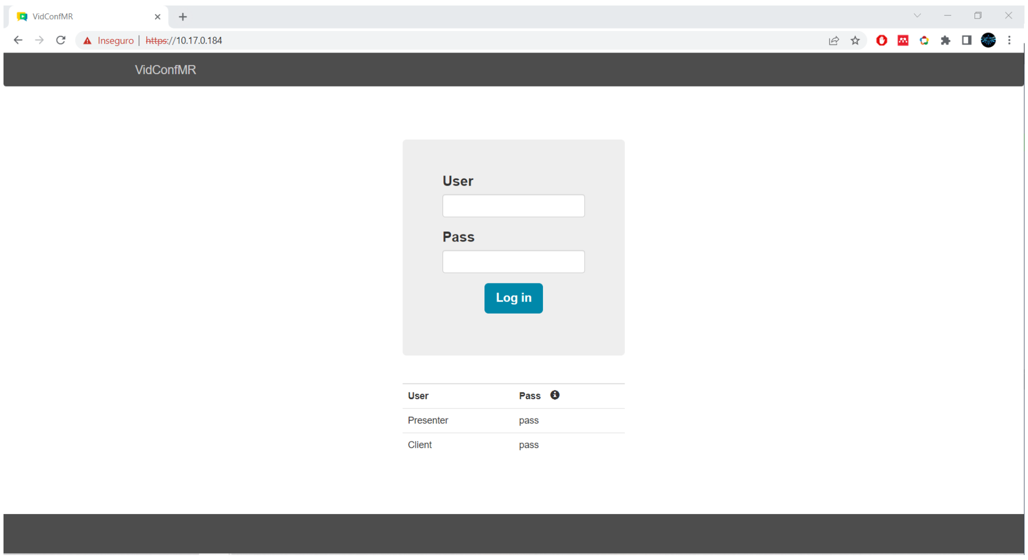
'ATENA' project info

Images section:

Figure 6.9: Example where the info panel for the fourth button is open.

6.7 End result

After describing everything done in order to create this application we will now take a look at the end result of our efforts. In figure 6.10 we can the entry page of the application. This is the result of the "index.html" file and is where the user can login into the application. For testing purposes there are two different login possibilities, one where the user is a presenter and another where the user is a visitor (there can be multiple people entering the sessions as visitors).



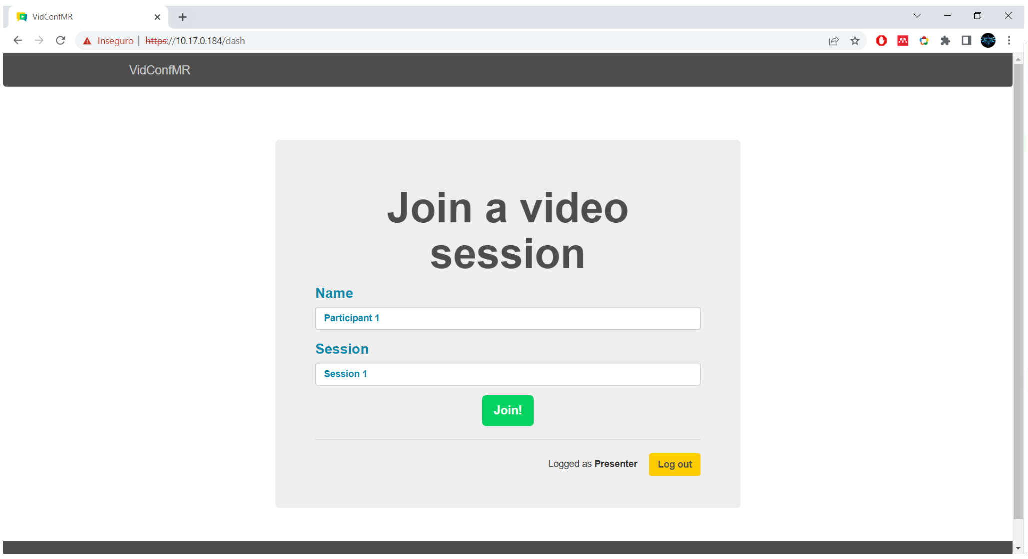
User

Pass

User	Pass
Presenter	pass
Client	pass

Figure 6.10: Entry point for the application where the users can login.

In figure 6.11 we can see the landing page after logging into the application. This is where the user can chose their nickname and what session they wish to join (for now all names are acceptable but there is a limit to the number of characters).



Join a video session

Name: Participant 1

Session: Session 1

Logged as **Presenter**

Figure 6.11: Page that users encounter after logging in, where they can chose the session.

After this the users will finally join a videoconferencing session where they will immediately see many of the implemented features. In figure 6.12 we can see a session where there is only one person present. We can see the big camera display that initially has the users video feed but can be changed after more users enter. There is also the top bar where we will be able to see all those who enter the session with their nicknames and a second tag that only appears if they are the presenter. We can see that this users nickname is "Participant 1" and that they are the presenter. It is also apparent that on the right side there is the chat functionality, on the bottom we have the mute mic and video buttons along side the leave session button and on the left we have an empty column that will have the extra information buttons.

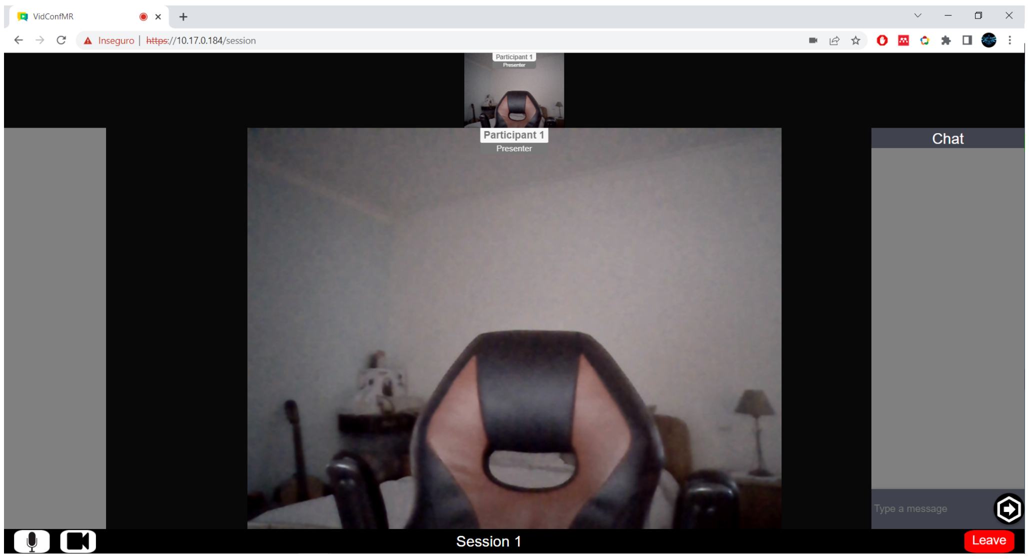


Figure 6.12: Example of a session with only the presenter inside.

We can then see the detection of the markers working by using one the predetermined barcode markers. Since we will be performing tests in the department where the information that is meant to be shown pertains to it, the example that can be seen in figure 6.13 has a model (which is a simple 3D tag) with a name of a section of the department. After the marker shows up on the camera feed a button is added to the left side column.

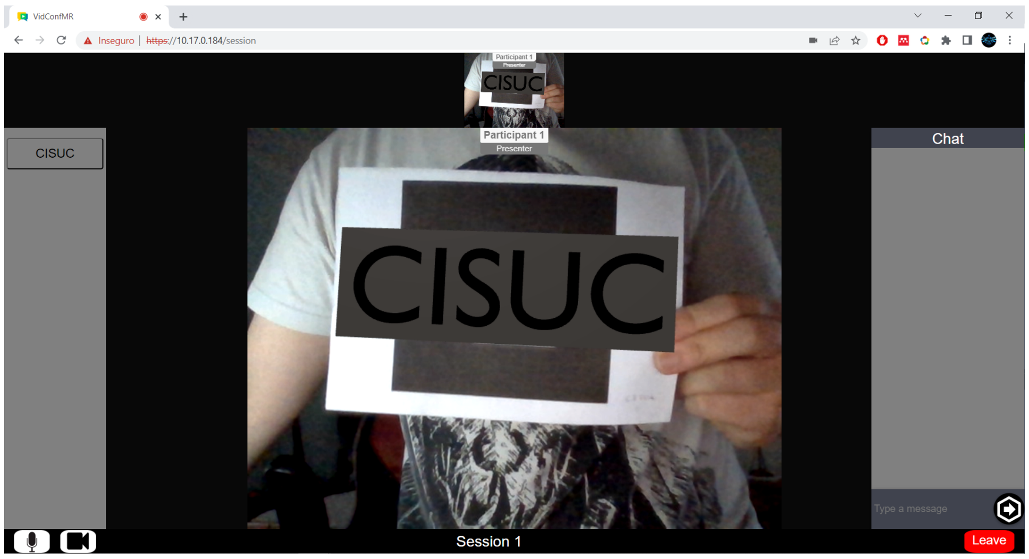


Figure 6.13: Example of a session with only the presenter detecting a marker.

In this next figure (fig 6.14 we can see that a new user has entered the session. Their nickname is "Participant 2" and since they do not have an extra tag they are a visitor. We can also see that both users have muted their mic as indicated by the symbol found in their camera feed on the top and by the fact that at bottom left of the page the symbol in the button has changed. Also some messages have been sent in the chat.

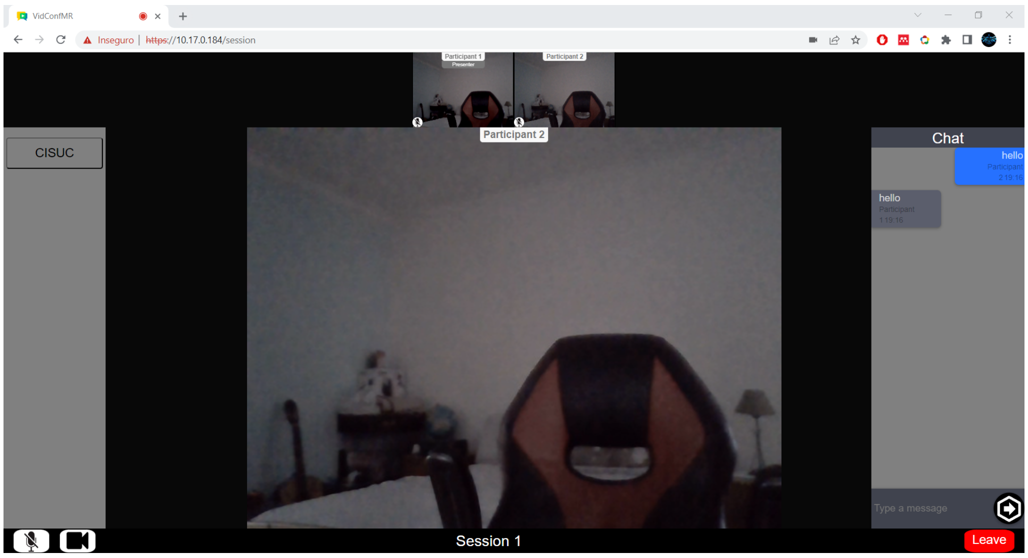
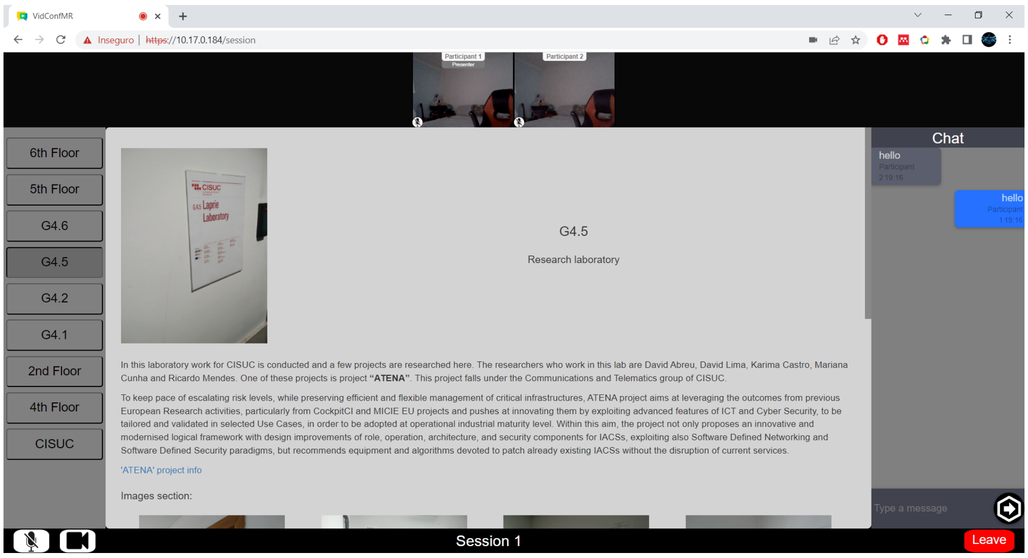


Figure 6.14: Example of a session with two users who have muted their mics and sent chat messages.

For our next example we have used the camera to detect many more marker to fill the left column with buttons. In figure 6.15 we are seeing the session through the eyes of a presenter that has pressed one of the buttons and is looking at the extra information that is displayed.



G4.5

Research laboratory

In this laboratory work for CISUC is conducted and a few projects are researched here. The researchers who work in this lab are David Abreu, David Lima, Karima Castro, Mariana Cunha and Ricardo Mendes. One of these projects is project **“ATENA”**. This project falls under the Communications and Telematics group of CISUC.

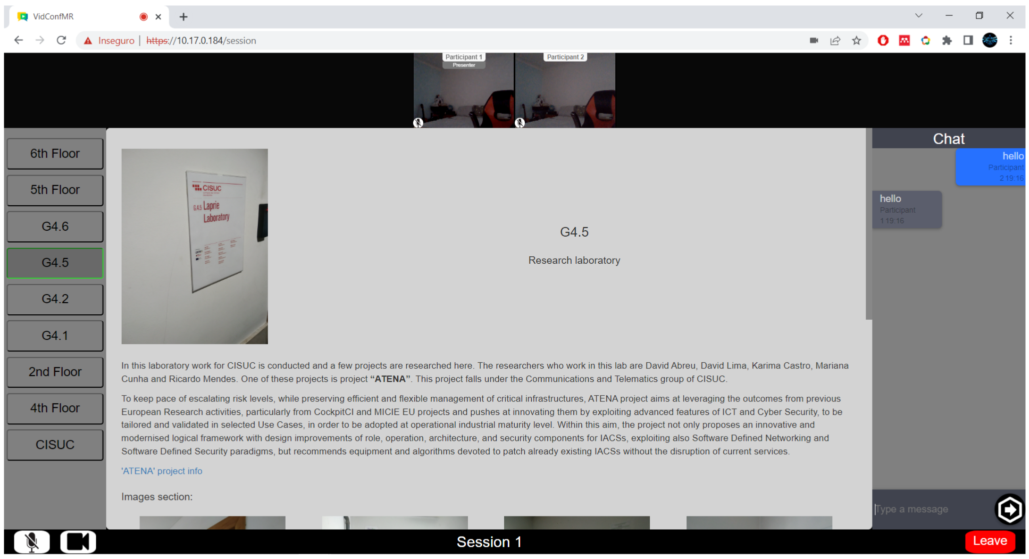
To keep pace of escalating risk levels, while preserving efficient and flexible management of critical infrastructures, ATENA project aims at leveraging the outcomes from previous European Research activities, particularly from CockpitCI and MICIE EU projects and pushes at innovating them by exploiting advanced features of ICT and Cyber Security, to be tailored and validated in selected Use Cases, in order to be adopted at operational industrial maturity level. Within this aim, the project not only proposes an innovative and modernised logical framework with design improvements of role, operation, architecture, and security components for IACSs, exploiting also Software Defined Networking and Software Defined Security paradigms, but recommends equipment and algorithms devoted to patch already existing IACSs without the disruption of current services.

'ATENA' project info

Images section:

Figure 6.15: Example of a session where the presenter has pressed one of the buttons.

In this final figure we intend to showcase that when the presenter opens one of the buttons that same button open in the client side while always having an outline over the button that the presenter is looking at. This can be seen in figure 6.16. It is also notable that the visitor can still look at other buttons while the presenter has a different one open (and the outline will remain in the one the presenter has open).



G4.5

Research laboratory

In this laboratory work for CISUC is conducted and a few projects are researched here. The researchers who work in this lab are David Abreu, David Lima, Karima Castro, Mariana Cunha and Ricardo Mendes. One of these projects is project **"ATENA"**. This project falls under the Communications and Telematics group of CISUC.

To keep pace of escalating risk levels, while preserving efficient and flexible management of critical infrastructures, ATENA project aims at leveraging the outcomes from previous European Research activities, particularly from CockpitCI and MICIE EU projects and pushes at innovating them by exploiting advanced features of ICT and Cyber Security, to be tailored and validated in selected Use Cases, in order to be adopted at operational industrial maturity level. Within this aim, the project not only proposes an innovative and modernised logical framework with design improvements of role, operation, architecture, and security components for IACSs, exploiting also Software Defined Networking and Software Defined Security paradigms, but recommends equipment and algorithms devoted to patch already existing IACSs without the disruption of current services.

'ATENA' project info

Images section:

Figure 6.16: Example of a session where the presenter has pressed one of the buttons from the visitor point of view.

With this we can see that the resulting application follows all of the established requirements in the previous chapter. All the functionalities mentioned in the requirements are present in the final version of the application.

6.8 Deployment

In order to have the application running in a way so that multiple people can access it through the internet it was necessary to deploy it in a server. Several options were considered for this, we looked at the possibility of using an AWS (Amazon Web Services) server to run our application or using Microsoft Azure. But ultimately this solutions required payment or did not allow for enough computer power to run the application. This application required at a minimum two CPUs, 8GB of RAM, about 30GB of space as well as a generous network bandwidth.

The solution that was used was to request the usage of a virtual machine inside the university's network to run the application. The university provided for this purpose a virtual machine with 4 CPUs, 16GB of RAM and 100GB of storage space (Figure 6.17). The only downside to this solution is that the application does not have a public address which means that it can only accessed when connected to the university's network. When stetting up the server it was important to set up the ports correctly. Some ports had to be opened to allow the media server to establish communications, others for TURN/STUN operations and for

OpenVidu connections. It was also necessary to set up a certificate to allow for communications. A self-signed certificate was used since we did not have a public address. This only leads to a warning page showing up for the users before connection to the application. After that is done anyone connect to the university's network can use a predetermined IP address to connect to the application. With this the application is up and running, ready to be used and tested.

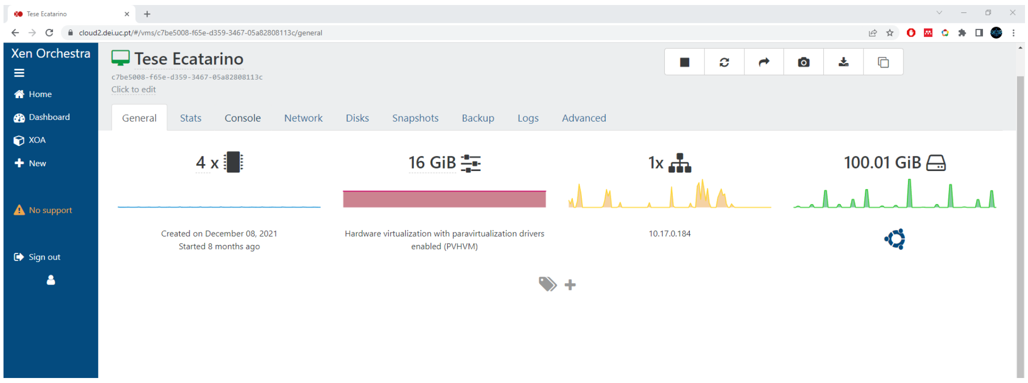

Tese Ecatarino
cloud2.dei.uc.pt/#/vms/c7be5008-f65e-d359-3467-05a82808113c/general
Xen Orchestra
Home
Dashboard
XOA
New
No support
Sign out
Tese Ecatarino
c7be5008-f65e-d359-3467-05a82808113c
Click to edit
General
Stats
Console
Network
Disks
Snapshots
Backup
Logs
Advanced
4 x
16 GiB
1x
100.01 GiB
Created on December 08, 2021
Started 8 months ago
Hardware virtualization with paravirtualization drivers enabled (PVHVM)
10.17.0.184

Figure 6.17: Web page to access the virtual machine provided by the university.

Chapter 7

Evaluation

With the application fully developed and its functionalities created it was necessary to validate all of its components. For this purpose several formal and informal tests (where the formal test are documented and the informal are not) were performed and its results taken into consideration. The first tests that were performed were to ensure that the application worked as expected on a variety of platforms and browsers alongside its functionalities. Next we tested the detection capabilities of the application in order to discover the best practices for creating and setting up markers in the chosen location for the presentation. In the end we performed usability tests with new testers who had never been in contact with the application to get an impartial view on its usability.

7.1 Functional Tests

In order to ensure that all the functionalities created were in working order it was necessary to test the application. The first round of test was done in a formal setting, with the test scenario previously prepared and planned. This test was performed in the university with the help of a colleague.

In this test six slightly different scenarios, where the application was used by at least one presenter and one visitor, were performed. During this six scenarios the main differences revolved around the type and number of devices used as well as the type of browser. While performing the tests a set of steps were followed to ensure that all the implemented functionalities were tested properly. All tests went as expected and afterwards all results were documented. The documentation of this test can be found in Appendix B.

The results of this test showed us that the application functions in both Windows computers and Android cellphone, using Chrome and/or Firefox as a browser. It also showed that with the exception of a few bugs all the tested functionalities worked as expected. It's important to note that all the bugs that were found were minimal in their impact to the functioning of the application, and that they were all fixed shortly after performing this test.

After performing this tests more informal tests were performed to ensure that all bugs had been fixed and to see how many more platforms and browser were supported by the application. The results showed us that the application does not function correctly in MacOS and iOS, but that it works on Windows and Android devices of any shape and size. For browsers the application works on Chrome, Firefox, Edge and Opera.

7.2 Marker Tests

The next round of tests were performed with the explicit intent of figuring out what would be the best practices for the creation of markers and where best to place them in the environment. We prepared six different testing scenarios that would ensure that all necessary facets of the placement of markers would be tested.

For this we had to compare a couple different types of markers, we compared the effect the size of the marker has, how much the light affects detection and at what angles does the application begin to detect the markers. For the first three we used the distance at which the application detected the markers in order to determine the best results. For the last one we used the angle from the wall to where the marker starts being detected, while maintaining the same distance from the marker.

Many of the variables used for each scenario came from the results of previous scenarios, for example after testing for the best type of marker, the one that was considered the best is the one used in the next scenarios. The order that the scenarios were performed was first the type of marker, next was the size of the marker, afterwards was the angle and finally the how the light affects the detection. To perform the tests several steps for each scenario were prepared to ensure the best results. Also worth mentioning that each scenario was done several times in a row for more accurate results. This test was done in a formal way at the university with the help of a colleague, where the tests done and results were documented. The documentation of this test can be found in Appendix C.

The results of the tests showed us that from the two type of markers that can be used in our application, "barcode" type (example in figure 7.1) and "custom" type (example in figure 7.2), the one that is detected from furthest away is the "barcode" type with a difference of about 2 meters. It also showed us that the size of the marker has a big effect on the distance at which the markers are detected. So the optimal size of marker will depend a lot on the specific environment used for the presentation. If the markers are intended to be placed somewhat away from the presenter it might be best to use A4 size as this allows for detection up to 4 meters away. While if the presenter can get closer to the markers a smaller size like A5 is sufficient at 2,5 meters of detection range.

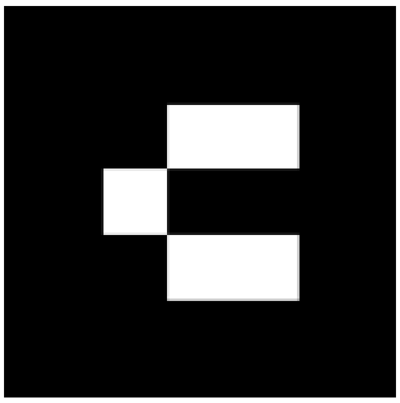


Figure 7.1: Example of a "barcode" type marker.

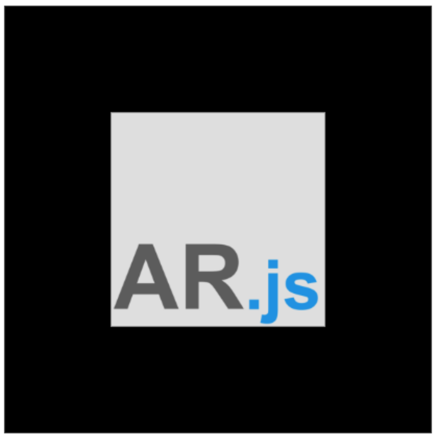


Figure 7.2: Example of a "custom" type marker.

The tests also showed us that the markers can be detected while looking at them at an average of 35 degrees from the wall it is placed on. This means that the presenter should, during a presentation, be able to look at the markers from at least 35 degrees from the wall. When it comes to lighting the tests showed us that the better lit the markers are the better they are detected. It is worth mentioning that for the markers we used simple paper, if the markers are printed on different materials there can be other problems to consider such as the reflection of the light making it difficult to detect the marker.

7.3 Usability Tests

With the application running as planned it was necessary to get outside opinions of it to validate its purpose. To this end a test was prepared where several people would try out the application for the very first time. With this in mind a presentation using the MR elements that the application provides was prepared. We decided to make a presentation of CISUC (Centre for Informatics and Systems of the University of Coimbra) laboratories in DEI (as seen in figure 7.3).

This would include the presenter walking through a predefined section of the university were several markers would be placed in strategic places where they would contain information regarding the several laboratories of CISUC (an example laboratory can be seen in figure 7.4). For this purpose ten "barcode" markers of size A5 where prepared, each one having a different 3D model and information associated with it. Each model was a simple tag with the name of what needed to be explained, for example each laboratory had a tag with the name of the room. And for each model a different button and panel with information was created and added to the application.

Figure 7.3: Hallway of CISUC where the tests took place.

Figure 7.4: Entrance to laboratory G6.2.

A plan of the presentation was made so the presenter would be able to do it by following a script, and also a small speech that explained the basic functioning of the application was prepared to be given to the presenter at the start of the tests. The presenter would be given an Android device with Chrome on it to perform the tests. For the visitors it would be asked they bring a laptop with Windows and one of the browsers that work with the application installed. It would also be asked to bring a pair of headphones, since they would be in the same room and it could create an echo. A small speech was also prepared to give at the beginning of the presentation for the visitors. For the end of the tests, to be able to gather as much information as possible from the users, a few question were prepared to incentivise a conversation where they could give their opinions. Also a questionnaire using the System Usability Scale (SUS) questions and a few custom ones would be given after.

With the presentation prepared a group of people was put together in the same room and each would try to use the application using the least amount of information possible, with one using the application as a presenter. All this process was documented and can be found in Appendix D.

The results of this tests showed us that for the most part the users found that the application is intuitive and easy to use. The application follows what is usually expected out of a videoconferencing application and adds the MR elements on top without it becoming confusing. The users also found that the way to add the MR elements is well done and that it adds to the presentation making it more interactive and informative. At the same time the users had a few issues with the application. Most found that the design of the application was not very appealing and that it should be improved. They also thought of a few changes that would improve the application such as adding the ability to zoom in pictures, to hide the chat window, the ability to resize the UI among others. There was also the idea of adding new ways to use the panels to increase interactivity like for example using the panel as a way of making question to the visitors and getting their responses. All in all the users seemed to enjoy using the application and were happy to comment on how it could be improved.

Chapter 8

Conclusion

With the conclusion of this project we have managed to create a prototype of an application that unites MR technology and videoconferencing technology in a useful and interesting way. We were able to combine these technologies into an interactive and immersive videoconferencing experience. With all the research done in both areas related to these technologies we found the best tools to create the application and found a way for them to work simultaneously.

While uniting these technologies we were able to do it in a way that neither of them lost any of their capabilities and functions. Also by encapsulating the technologies into a web page we managed to remove the need for an installation and allowed for it to be accessible to many people. We also have the advantage of being able to use all of the available resources on the internet when creating a web page. Creating the web page from scratch gave us the possibility to create the application with the design and all the functionalities required.

Having this application as an option to make a presentation will hopefully add a new way of showcasing information in a fun and immersive way. It is possible to use this application to create many presentations that can be more interesting and creative than simply sharing a PowerPoint presentation for example. The application allows for demonstrating how anything works by showing it working live with videoconferencing but still adding any extra information by using MR.

There are however still a few ways to improve on this project. It was commented during the tests performed to validate the application that the design and UI can still be made better, while all planned functionalities were implemented in this project, some might not have been added in the best of ways. Currently all information and models shown in the application were added directly into it, a good improvement would be to create a database were these variable could be easily changed (most of these variables were already added to the application in such a way where making this change would not to be to complicated). An interesting idea given by a few users that tested this application was to add new interaction to the application. What this means is that alongside having extra information shown in the panels we could have other things like questionnaires for example.

All in all, I believe that the project created managed to fulfill its objective by creating an application that can be used to enhance an activity that is quite common in today's world. By creating and testing the application we proved that it has potential in its idea and execution, and that with some improvements it can be easily used by whomever needs it.

References

[1] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," *Telemanipulator and Telepresence Technologies*, vol. 2351, pp. 282–292, 1995. [Online]. Available: http://vered.rose.utoronto.ca

[2] D. Barnard, "History of VR - Timeline of Events and Tech Development," 2019. [Online]. Available: https://virtualspeech.com/blog/history-of-vr

[3] B. Poetker, "A Brief History of Augmented Reality (+Future Trends & Impact)," 2019. [Online]. Available: https://www.g2.com/articles/history-of-augmented-reality

[4] I. Mustaqim, S. T. Pd, and N. Kurniawan, "Augmented Reality," ICCUBEA, Tech. Rep. 7, jul 2016. [Online]. Available: www.ijert.org

[5] roOomy B.V., "3D, Augmented and Virtual Reality Interior Design Apps | roOomy," 2021. [Online]. Available: https://rooomy.com/interior-design

[6] Niantic, "Niantic Support," 2020. [Online]. Available: https://niantic.helpshift.com/a/pokemon-go/?l=en&p=web&s=finding-evolving-hatching&f=catching-pokemon-in-ar-mode

[7] W. R. Sherman and A. B. Craig, *Understanding Virtual Reality*. Elsevier, 2019.

[8] T. News, D. C. Washington, and D. C. June, "Embry-Riddle Aeronautical University : New Virtual Reality Flight Simulator Offers Glimpse of Future Training Tool," pp. 1–2, 2020. [Online]. Available: https://news.erau.edu/headlines/new-virtual-reality-flight-simulator-offers-glimpse-of-future-training-tool

[9] K. Ugolik, "A Look Into Virtual Reality Therapy - Is This the Future of Therapy? | Freethink," 2019. [Online]. Available: https://www.freethink.com/articles/is-the-future-of-therapy-virtual-a-look-into-virtual-reality-therapy

[10] H. Kato and M. Billinghurst, "A mixed reality 3D conferencing application," Human Interface, Tech. Rep. May 2014, 1999. [Online]. Available: http://www.hitl.washington.edu/publications/r-99-1/

[11] A. B. Craig, *Understanding Augmented Reality*. Elsevier, 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/C20110072496

[12] R. Paul, "The Best Car GPS | Reviews by Wirecutter," 2019. [Online]. Available: https://www.nytimes.com/wirecutter/reviews/best-car-gps/

[13] F. Zhou, H. B. L. Dun, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," in *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008*, 2008, pp. 193–202.

[14] E. Bostanci, N. Kanwal, S. Ehsan, and A. F. Clark, "User Tracking Methods for Augmented Reality," *International Journal of Computer Theory and Engineering*, pp. 93–98, 2013. [Online]. Available: https://www.researchgate.net/publication/259936367_User_Tracking_Methods_for_Augmented_Reality

[15] Microsoft, "Microsoft HoloLens | The leader in mixed reality technology," 2018. [Online]. Available: https://www.microsoft.com/en-us/hololens

[16] Wikitude, "Extended Tracking augments reality beyond target images." 2021. [Online]. Available: https://www.wikitude.com/augmented-reality-extended-tracking/

[17] M. Klopschitz, G. Schall, D. Schmalstieg, and G. Reitmayr, "Visual tracking for augmented reality," IPIN, Tech. Rep., 2010.

[18] Shopify, "Shopify AR powered by 3D Warehouse · Shopify Help Center," 2021. [Online]. Available: https://help.shopify.com/en/manual/products/product-media/3d-warehouse

[19] A. Ables, "Augmented and virtual reality: Discovering their uses in natural science classrooms and beyond," pp. 61–65, 2017. [Online]. Available: https://ardev.es/en/augmented-reality/

[20] K. Barranco, "S.L.A.M — Tracking for MR. S.L.A.M. is an acronym used in the... | by Kade Barranco | desn325-EmergentDesign | Medium," 2019. [Online]. Available: https://medium.com/desn325-emergentdesign/s-l-a-m-tracking-for-mr-2a0face9d97a

[21] S. Liddell, "Locatify - Location Based Augmented Reality Apps in 2017 (RTLS amp; AR)," 2017. [Online]. Available: https://locatify.com/blog/location-based-augmented-reality-apps-2017-rtls-ar/

[22] T. English, "How Do VR Headsets Work?" 2020. [Online]. Available: https://interestingengineering.com/vr-headsets-work-through-a-combination-of-different-tracking-technologies

[23] F. Cutolo, N. Cattari, U. Fontana, and V. Ferrari, "Optical See-Through Head-Mounted Displays With Short Focal Distance: Conditions for Mitigating Parallax-Related Registration Error," *Frontiers in Robotics and AI*, vol. 7, p. 572001, dec 2020. [Online]. Available: www.frontiersin.org

[24] A. Charlton, "From Oculus to Valve, these are the top gaming VR headsets - Gearbrain," 2020. [Online]. Available: https://www.gearbrain.com/best-vr-headsets-for-gaming-1979184681.html

[25] Wikimedia, "Optical head-mounted display - Wikipedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Optical_head-mounted_display

[26] M. R. Mine, J. Van Baar, A. Grundhöfer, D. Rose, and B. Yang, "Projection-based augmented reality in Disney theme parks," *Computer*, vol. 45, no. 7, pp. 32–40, 2012.

[27] V. Ilyukha, "10 Best Augmented Reality App Development Tools," 2021. [Online]. Available: https://jelvix.com/blog/5-best-tools-for-ar-development

[28] Apple, "ARKit - Augmented Reality - Apple Developer," 2021. [Online]. Available: https://developer.apple.com/augmented-reality/arkit/

[29] Google: ARCore, "Developers ARCore Overview," 2020. [Online]. Available: https://developers.google.com/ar/discover/

[30] ARToolKit, "ARToolKit Home Page," 2021. [Online]. Available: http://www.hitl.washington.edu/artoolkit/

[31] ——, "ARToolKit licensing," 2021. [Online]. Available: http://www.hitl.washington.edu/artoolkit/license.html

[32] HITLab, "ARToolKit Documentation (How does ARToolKit work?)," 2006. [Online]. Available: http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm

[33] AR.js, "AR.js Documentation," 2021. [Online]. Available: https://ar-js-org.github.io/AR.js-Docs/

[34] Wikitude GmbH, "Wikitude Augmented Reality: the World's Leading Cross-Platform AR SDK," 2020. [Online]. Available: https://www.wikitude.com

[35] Wikitude, "Augmented Reality Experiences with Wikitude Cross Platform SDK," 2021. [Online]. Available: https://www.wikitude.com/products/wikitude-sdk/

[36] ——, "Wikitude Store: Find Best Pricing for your Augmented Reality Experiences." 2020. [Online]. Available: https://www.wikitude.com/store/

[37] PTC, "Getting Started | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/

[38] Wikimedia, "PTC (software company) - Wikipedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/PTC_(software_company)

[39] Vuforia, "Vuforia Engine | Create AR Apps and AR Experiences | PTC," 2021. [Online]. Available: https://www.ptc.com/en/products/vuforia/vuforia-engine

[40] ——, "vuforia. Vuforia Engine Overview VuforiaLibrary," 2020. [Online]. Available: https://library.vuforia.com/features/overview.html

[41] PTC, "Tools Overview | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/tools/overview.html

[42] ——, "AR App Development With Vuforia Engine | PTC," 2021. [Online]. Available: https://www.ptc.com/en/products/vuforia/vuforia-engine/ar-app-development

[43] ——, "Usage | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/articles/FAQ/Usage.html

[44] Vuforia, "Vuforia Engine pricing | PTC," 2020. [Online]. Available: https://www.ptc.com/en/products/vuforia/vuforia-engine/pricing

[45] WebRTC, "WebRTC," 2021. [Online]. Available: https://webrtc.org/

[46] Google, "WebRTC," 2021. [Online]. Available: https://webrtc.org/

[47] R. Power, "WebRTC for New Users: Terminology and How It Works | Wowza," 2020. [Online]. Available: https://www.wowza.com/blog/webrtc-terminology-and-how-it-works

[48] Kurento, "Kurento," 2014. [Online]. Available: https://www.kurento.org/

[49] ——, "Kurento," 2014. [Online]. Available: https://www.kurento.org/

[50] OpenVidu, "OpenVidu," 2020. [Online]. Available: https://openvidu.io/

[51] Vuforia, "Model Targets Native Workflow | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/articles/Solution/model-targets-workflow.html

[52] ——, "Model Target Generator User Guide | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/articles/Solution/model-target-generator-user-guide.html

[53] ——, "Vuforia Area Target Creator App | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/features/environments/area-targets/creator-app.html

[54] ——, "Best Practices for Preparing and Scanning an Environment | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/features/environments/area-targets/best-practices-for-preparing-and-scanning-an-environment.html

[55] ——, "Introduction to Ground Plane in Unity | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/articles/Solution/ground-plane-guide.html

[56] Microsoft, "About Unity Render Streaming | Unity Render Streaming | 3.1.0-exp.3," 2020. [Online]. Available: https://docs.unity3d.com/Packages/com.unity.renderstreaming@3.1/manual/index.html

[57] Unity Technologies, "Index | MixedReality-WebRTC Documentation," 2022. [Online]. Available: https://microsoft.github.io/MixedReality-WebRTC/index.html

[58] Vuforia, "VuMark | VuforiaLibrary," 2021. [Online]. Available: https://library.vuforia.com/features/objects/vumark.html

Appendices

Appendix A

Mockups

This Appendix holds all of the mockups created for this application. They were created to guide the design of the application throughout the project. There were three different versions of the mockups that were created and changed as it was deemed necessary. Bellow you will find all the mockups in the order they were made.

A.1 Version 1

In this early version the design was meant to be simple, with most of the screen being the camera feed (the white background with a cross). At this time we also had a different design for the presenter and the visitor since the presenter would most likely be used in a mobile device. During this time we also not yet planed to make a web page so the design did not reflect that choice.

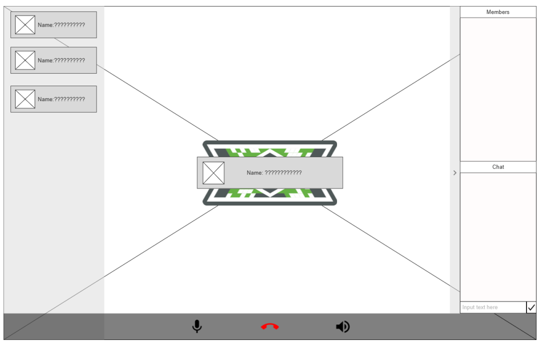


Figure A.1: Mockup of the visitor side of the application.

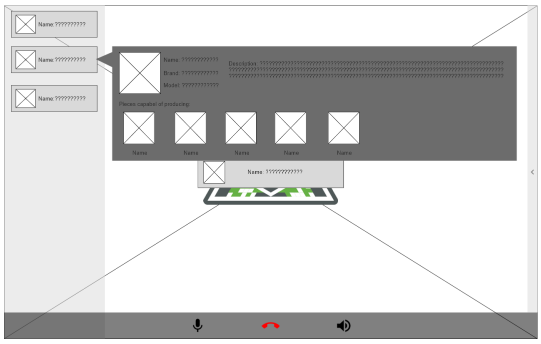


Figure A.2: Mockup of the visitor side of the application with extra information.

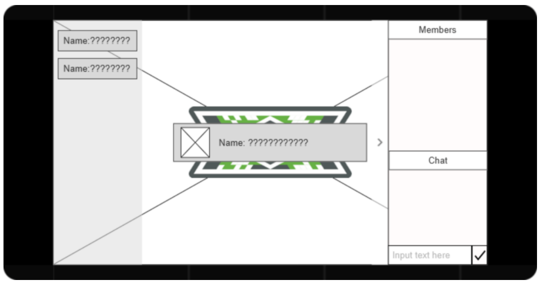


Figure A.3: Mockup of the presenter side of the application.

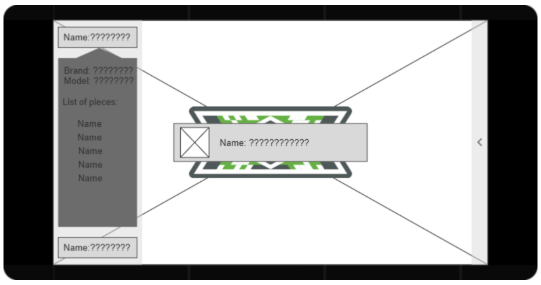


Figure A.4: Mockup of the presenter side of the application with extra information.

A.2 Version 2

For the second version of the mockups we cleaned the design to look better. We also added the ability to see everyone camera feed, made the information panel bigger and adjusted the design to fit with a web page.

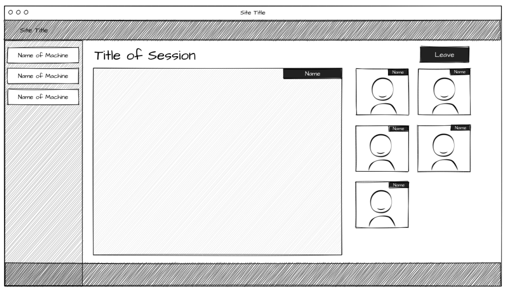


Figure A.5: Second mockup of the visitor side of the application.

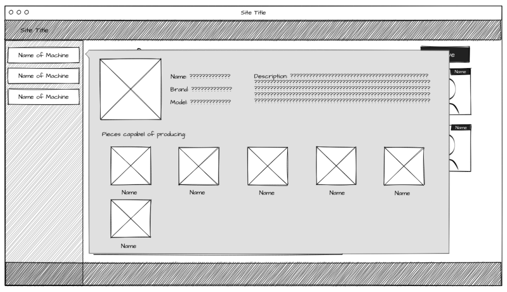


Figure A.6: Second mockup of the visitor side of the application with extra information.

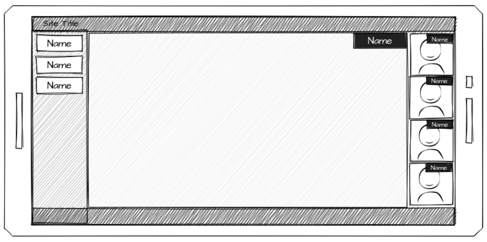


Figure A.7: Second mockup of the presenter side of the application.

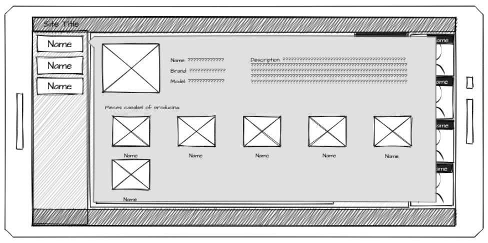


Figure A.8: Second mockup of the presenter side of the application with extra information.

A.3 Version 3

For the final version off the mockups we changed the design to add all the new planned features. We also made it look more like a videoconferencing application. Since it was a web page there was no need to differentiate so a change was made to make the presenter and visitor side look the same, while still working in slight different ways.

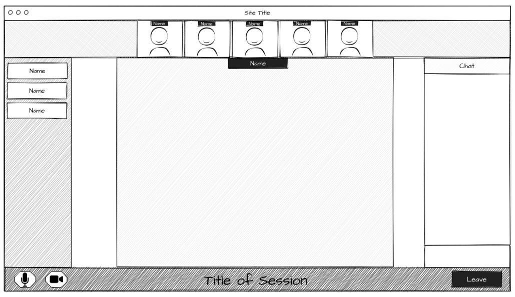


Figure A.9: Third mockup of the application.

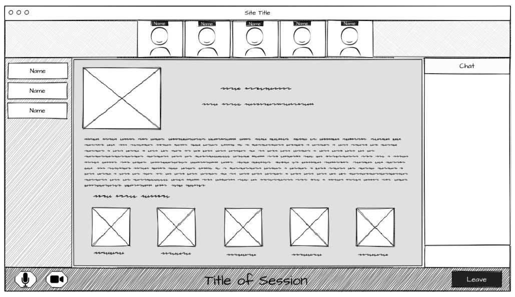


Figure A.10: Third mockup of the application with extra information.

Appendix B

Functional tests

In this Appendix you will find the documentation created for the functional tests on the application. This includes the preparation and script for the test as well as the results.

Introduction

In this phase of the project, it was necessary to understand how well the application managed to perform in a more realistic scenario using several types of devices and web browsers. It was also important to verify if all the implemented functionalities of the application worked according to specification.

To this end several tests were conducted that allowed us to have a better understanding of the application's overall performance. In these tests we used 4 devices, an android that would function as the presenter, another android that would function as a client and two more clients on two windows PC. The devices also had installed two different web browsers (Chrome and Firefox) to determine if there were any problems. Also, three different markers were prepared and printed.

Tests

[TEST1]

- Description:
  - Check performance of the presenter side of the application in Chrome.
  - Check marker detection, popup operation, stream speed, audio mute feature and video mute feature.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Chrome on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Press “Mute” button.
  - Press “Unmute” button.
  - Press “Turn off Camera” button.
  - Press “Turn on Camera” button.
  - Aim camera at marker 0.
  - Aim camera at marker 1.
  - Aim camera at marker 2.
  - Open marker 0 popup.
  - Open marker 1 popup.
  - Open marker 2 popup.
  - Close marker 2 popup.

Results:

The test was successfully performed with some problems detected. These problems were: Model 1 does not appear on the marker, when opening a second popup the previous one remains open until closed (to have a more user-friendly system when opening a popup another open one must close), model not centred on the marker, username tag is out of position and slow stream speed.

[TEST2]

- Description:
  - Check performance of the application's presenter side in Firefox.
  - Check marker detection, popup operation, stream speed, audio mute feature and video mute feature.
  - Assumption:

  - Application built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Firefox on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Press "Mute" button.
  - Press "Unmute" button.
  - Press "Turn off Camera" button.
  - Press "Turn on Camera" button.
  - Aim camera at marker 0.
  - Aim camera at marker 1.
  - Aim camera at marker 2.
  - Open marker 0 popup.
  - Open marker 1 popup.
  - Open marker 2 popup.
  - Close marker 2 popup.

Results:

The results were the same as [TEST1]. The test was successfully performed with some problems detected. These problems were: Model 1 does not appear on the marker, when opening a second popup the previous one remains open until closed (to have a more user-friendly system when opening a popup another open one must close), model not centred on the marker, username tag is out of position and slow stream speed.

[TEST3]

- Description:
  - Check the performance of the application in Chrome with a presenter and 3 clients (2 on Windows and 1 on Android).
  - Check client detection, client popups, presenter control popups, stream speed, mute client audio, turn off client video.
- Assumption:
  - Application built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Chrome on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open Chrome on 2 Windows devices and 1 on android.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Press "Mute" button on clients.
  - Press "Unmute" button on clients.
  - Press "Turn off Camera" button on clients.
  - Press "Turn on Camera" button on clients.
  - Point presenter camera at marker 0.
  - Point presenter camera at marker 1.
  - Point presenter camera at marker 2.
  - Open marker 0 popup on client.
  - Open marker 1 popup on client.
  - Open marker 2 popup on client.
  - Close marker 2 popup on client.
  - Open marker 0 popup in presenter.
  - Open marker 1 popup in presenter.
  - Open marker 2 popup in presenter.
  - Close marker 2 popup in presenter.

Results:

The test was successfully performed with some problems detected. These problems were: Android video window size is different from Windows, android client video does not rotate to force view in "landscape" mode and presenter does not appear on big screen right away.

[TEST4]

- Description:
  - Check the performance of the application in Firefox with a presenter and 3 clients (2 on Windows and 1 on Android).
  - Check client detection, client popups, presenter control popups, stream speed, mute client audio, turn off client video.
- Assumption:
  - Application built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Firefox on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open Firefox on 2 Windows devices and 1 on Android.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Press “Mute” button on clients.
  - Press “Unmute” button on clients.
  - Press “Turn off Camera” button on clients.
  - Press “Turn on Camera” button on clients.
  - Point presenter camera at marker 0.
  - Point presenter camera at marker 1.
  - Point presenter camera at marker 2.
  - Open marker 0 popup on client.
  - Open marker 1 popup on client.
  - Open marker 2 popup on client.
  - Close marker 2 popup on client.
  - Open marker 0 popup in presenter.
  - Open marker 1 popup in presenter.
  - Open marker 2 popup in presenter.
  - Close marker 2 popup in presenter.

Results:

The results were the same as for [TEST3]. The test was successfully performed with some problems detected. These problems were: Android video window size is different from Windows, android client video does not rotate to force view in “landscape” mode and presenter does not appear on big screen right away.

[TEST5]

- Description:
  - Check application performance in Chrome and Firefox with a presenter in Chrome and 3 clients (2 in Windows and 1 in Android) with 2 in Firefox and 1 in Chrome.
  - Check client detection, client popups, presenter control popups, stream speed, mute audio on client, turn off video on client, check connection between different web browsers.
- Assumption:
  - Application built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Chrome on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open Firefox on 1 Windows device, Chrome on 1 Windows device and Firefox on 1 android device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Press "Mute" button on clients.
  - Press "Unmute" button on clients.
  - Press "Turn off Camera" button on clients.
  - Press "Turn on Camera" button on clients.
  - Point presenter camera at marker 0.
  - Point presenter camera at marker 1.
  - Point presenter camera at marker 2.
  - Open marker 0 popup on client.
  - Open marker 1 popup on client.
  - Open marker 2 popup on client.
  - Close marker 2 popup on client.
  - Open marker 0 popup in presenter.
  - Open marker 1 popup in presenter.

- Open marker 2 popup in presenter.
- Close marker 2 popup in presenter.

Results:

The results were the same as for [TEST3]. The test was successfully performed with some problems detected. These problems were: Android video window size is different from Windows, android client video does not rotate to force view in "landscape" mode and presenter does not appear on big screen right away.

[TEST6]

- Description:
  - Check application performance in Chrome and Firefox with a presenter in Firefox and 3 clients (2 in Windows and 1 in Android) with 2 in Chrome and 1 in Firefox.
  - Check client detection, client popups, presenter control popups, stream speed, mute audio on client, turn off video on client, check connection between different web browsers.
- Assumption:
  - Application built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open Firefox on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open Chrome on 1 Windows device, Firefox on 1 Windows device and Chrome on 1 android device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Press "Mute" button on clients.
  - Press "Unmute" button on clients.
  - Press "Turn off Camera" button on clients.
  - Press "Turn on Camera" button on clients.
  - Point presenter camera at marker 0.
  - Point presenter camera at marker 1.
  - Point presenter camera at marker 2.
  - Open marker 0 popup on client.
  - Open marker 1 popup on client.
  - Open marker 2 popup on client.
  - Close marker 2 popup on client.
  - Open marker 0 popup in presenter.
  - Open marker 1 popup in presenter.

  - Open marker 2 popup in presenter.
  - Close marker 2 popup in presenter.

Results:

The results were the same as for [TEST3]. The test was successfully performed with some problems detected. These problems were: Android video window size is different from Windows, android client video does not rotate to force view in "landscape" mode and presenter does not appear on big screen right away.

Conclusion

With the result from these tests, we can see that application for the most part function as expected with most of the functionalities implemented working correctly. But while the application can function it still has a few bugs that need to be fixed in order for the users to have a good experience with it.

Appendix C

Marker tests

In this Appendix you will find the documentation created for the marker tests on the application. This includes the preparation and script for the test as well as the results.

Introduction

In this phase of the project, it was necessary to understand how the application would respond to the markers depending on their characteristics and their position in the environment. This information is very important so the location where the application will be used can be prepared accordingly, and the presenter can also know what to expect while using the application.

To this end several tests were conducted that allowed us to have a better understanding of how the application responds to the markers. In these tests we used two devices, an android to be used as a presenter and a windows device as a visitor. The tests were done in the laboratories of CISUC (Centre for Informatics and Systems of the University of Coimbra) as it is the planned location for further tests (the use case we plan to test is a presentation of the laboratories). Several types of markers with different sizes were printed for the tests. A laser distance meter was used to record the distances at which the markers were detected and to know the angles a protractor was used.

Tests

[TEST1]

- Description:
  - Check the detection distance of the marker from the application.
  - This marker is of “barcode” type and is size A4.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at a long distance from the marker.
  - Aim the camera directly at the marker.
  - Slowly walk towards the marker.
  - Stop when marker is detected.
  - Write down the distance at which the marker was detected.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST1	3,225m	3,674m	4,340m	4,624m	4,674m	4,1074m	0,634129m

[TEST2]

- Description:
  - Check the detection distance of the marker from the application.
  - This marker is of "custom" type and is size A4.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at a long distance from the marker.
  - Aim the camera directly at the marker.
  - Slowly walk towards the marker.
  - Stop when marker is detected.
  - Write down the distance at which the marker was detected.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST2	2,307m	2,262m	1,896m	1,789m	2,152m	2,0812m	0,228236m

[TEST3]

- Description:
  - Check the detection distance of the marker from the application.
  - This marker is of "barcode" type and is size A5.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at a long distance from the marker.
  - Aim the camera directly at the marker.
  - Slowly walk towards the marker.
  - Stop when marker is detected.
  - Write down the distance at which the marker was detected.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST3	2,904m	2,141m	2,753m	2,449m	2,402m	2,5298m	0,301609m

[TEST4]

- Description:
  - Check the detection distance of the marker from the application.
  - This marker is of “barcode” type and is size A6.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at a long distance from the marker.
  - Aim the camera directly at the marker.
  - Slowly walk towards the marker.
  - Stop when marker is detected.
  - Write down the distance at which the marker was detected.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST4	1,685m	1,922m	1,670m	1,799m	2,038m	1,8228m	0,157365m

[TEST5]

- Description:
  - Check the angle at which the marker will be detected by the application.
  - This marker is of “barcode” type and is size A4.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at an acceptable distance from the marker (1,5m) while close to the wall the marker is on.
  - Aim the camera at the marker.
  - Slowly walk to face the marker while keeping the same distance from it.
  - Stop when marker is detected.
  - Write down the angle at which the marker was detected.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST5	40°	35°	30°	30°	35°	34°	4,1833°

[TEST6]

- Description:
  - Check the difference in distance at which the marker will be detected by the application when the marker is placed in high and low light.
  - This marker is of “barcode” type and is size A4.
- Assumption:
  - Application is built.
  - Application running on server.
  - Printed marker.
  - Marker added in the application.
- Steps:
  - Open web page on an Android.
  - Enter IP 10.17.0.184.
  - Log in as presenter.
  - Enter session 0.
  - Open web page for client on different device.
  - Enter IP 10.17.0.184.
  - Log in as customers.
  - Enter session 0.
  - Stand at a long distance from the marker in high light.
  - Aim the camera directly at the marker.
  - Slowly walk towards the marker.
  - Stop when marker is detected.
  - Write down the distance at which the marker was detected.
  - Repeat for the marker in low light.
  - Write down the difference in distance.
  - Repeat 5 times.

Results:

The test was successfully performed. The results were:

	1	2	3	4	5	Average	Standard deviation
TEST6	0,496m	0,564m	0,403m	0,319m	0,213m	0,3399m	0,139325m

(These results represent high light minus low light, therefor the marker in high light was always detected at a longer distance than the one in low light)

Conclusion

Given the results from the tests it is possible to ascertain that for must scenarios the usage of the “barcode” type markers will always be preferred for their easy detection. When talking about size it very much depends on the use case you need, in our case of making a presentation in the laboratories of CISUC a A5 size marker seems to be sufficient. The rest of the tests proved that the marker can be detected from a good angle and that the more light is hitting the marker the easier it is to detect.

Appendix D

Usability tests

In this Appendix you will find the documentation created for the usability tests on the application. This includes the preparation and script for the test as well as the results.

Introduction

This document was created with the purpose of organizing and describing the evaluation that will be performed by a group of people who haven't had contact with the application, to validate the implementation and functionalities of said application.

In the following sections the requirements of the application that are meant to be tested will be listed, a description of the test will be made, the necessary people, infrastructure and tools will also be described and finally the results of the test will be noted.

Requirements

In this project, a web application was created that allows for videoconferencing while adding Mixed Reality elements. This application will be used by a presenter and other users that will watch the presentation. There are several basic requirements that the application needs to meet to provide an acceptable experience to its users.

As a web application it needs to be accessible, its user interface should be **easily understood** making sure the users recognize what needs to be done at each step of its use. The application should also have **good performance** providing a smooth experience and ensure the users **security**. The application should be **accessible through a link**, should have a login for the presenter and the clients and should also accommodate for **several sessions** at once.

With the application being used for videoconferencing there are several functionalities that need to work properly. The video and audio transmissions should have **acceptable quality** and the users should have **control** over these transmissions (mute/unmute video and/or audio). There should also be a chat functionality so the users can communicate without using a mic.

All other functionalities that the application provides that are related to it using Mixed Reality must also be validated. The used **markers need to be recognized** by the application at acceptable distances, the 3D models need to show on top of the markers when recognized, the extra information buttons need to appear when the corresponding marker is recognized, the usage of the extra information should be easy, and the presenter should have extra control over the information that shows.

Test strategy

For this test we intend to simplify the use case into something that can be done and tested in the department. This way, all the requirements can be validated without having to involve and trouble more people than necessary.

The idea is to have someone give a presentation to a group of people of a small section of the department. In this part of the department several key features will be marked so the presenter knows to focus on them and these markers will add the Mixed Reality elements to the videoconference. Before the presentation starts, the use of the application will be explained to the presenter as they should understand its basic functionalities. At this time access to the application will be given to the users with enough information to enter the session but making sure not to explain too much so the user interface can be tested for its accessibility. At the beginning of the presentation the presenter will explain how the Mixed Reality features work so the users are not confused. During the presentation the presenter will walk through a designated path while focusing on the markers and giving the necessary information using the Mixed Reality functionalities of the application. After the presentation concludes all the users (including the presenter) will be given a questionnaire in order to obtain several statistics and opinions of the application, this questionnaire will be provided using a link.

People, tools and infrastructure

For this test we will require one person to act as the presenter, this person needs to have some knowledge of the application which includes some of the functionalities it provides, the presenter should also be aware of the path to take in the presentation alongside some information of the location to be able to give the presentation. There will also be several others to act as participants of the videoconference (at least three) whose task is to use the application to the best of their abilities. All the people involved should have some experience using other videoconferencing tools.

In order to join the videoconferencing session, the presenter needs to enter specific credentials so the application knows they are a presenter, the participants will have credentials specific for them. At the start of the test both the presenter and the participants will be given a link in order to enter the application easily. The link will be sent using email.

The presenter needs to use a mobile device for the presentation, this device must be an Android and needs to use an acceptable browser (Chrome, Firefox or Opera). The participants should each have a different computer with windows while using an acceptable browser (Chrome, Firefox, Edge or opera).

As for the location of the test, it was decided that the CISUC (Centre for Informatics and Systems of the University of Coimbra) laboratories in DEI could be used for the use case. This part of the department consists of a tower that spans six floors that house several laboratories used by CISUC with connections to other parts of the department such as classrooms and offices, special attention will be given to some laboratories on the 6th floor. Therefor a different marker will be printed and posted in each floor with some more markers in some laboratories on the 6th floor. This way the presenter walks up the tower stopping at each floor to give a small presentation and explanation of that floor, this explanation should be done with the use of the markers and the Mixed Reality elements.

Initial presentation

General presentation:

"Good afternoon, today you are here to test a videoconferencing application on the Web that aims to help make presentations using mixed reality by adding extra information that can be consulted. For this you will be given access to the application through a link sent by email. Then you are supposed to try to use the application yourself to see if it is accessible. In this test you will see a presentation of a guided tour of the department's CISUC tower with the presenter also testing the application. Therefore, the focus of this test is not the presentation itself and its content, but whether the application improves or worsens the delivery of information that will be given. There will always be someone with you in case something happens that you ca not solve on your own. At the end, when the presentation is over and the presenter returns, we will have a short conversation about the application followed by a questionnaire that will be sent to you by email. With that, this testing session will end.”

Extra for presenter:

“As a presenter you will have access to the application through a specific account. What you need to do is, as you go through each floor of the tower, you must first look at the marker found there until the 3D model appears or a button on the left side bar, on the top floor you must walk along the corridor and look at the markers next to the rooms. This button will contain extra information about the room and when you load the information it occupies the main video on your side (you can still see your video on the top bar). You also have to be careful that as a presenter when you press the button users will be notified that you are seeing that button, what happens is that the same button on the visitors side has a green border and if they do not have anything open it will open on the visitor side as well. At the beginning of your presentation, in the first marker, inform the visitors that the buttons contain information about what you are going to say and that when you open them, a green border appears.”

Script

The presentation will begin at the CISUC bar on the ground floor of the Department of Informatics Engineering (DEI). The presenter should go to the elevator area of the G tower of the DEI. Here the presenter will find the first marker on the right side of the elevator. The presenter should point the camera of the mobile device at the marker until the model that says "CISUC" appears.

At this point, a brief explanation of how the buttons added by bookmarks work should be given (ex: "During this presentation I will go through several bookmarks and whenever I pass by one a button will appear on the left side of the application that you can use to have extra information about the subject I will talk about. Also important to mention that when I open a button it will also open on your side unless you are seeing another button, anyway there will always be a green border on the button that I will be talking about.").

Then you should talk a little about the floor you are on and introduce CISUC (ex: "CISUC or Center for Informatics and Systems of the University of Coimbra is a research center that is currently working on a wide variety of projects, some which we will see today. CISUC occupies tower G of the DEI, which we will visit floor by floor. Here on the ground floor we have the CISUC bar with access to the entrance of the department and access to the rest of the tower.") (This information will be available on the application buttons).

After this introduction you will go up to the second floor where you will find a marker on the right side of the elevator. Then you should show the place and talk a little about the floor. (example: "On this floor we have access to the offices of several DEI professors and access to laboratories F2.1 to F2.3. In these laboratories we can find the AC team, Adaptive Computation, from CISUC") (This information will be available on the buttons on the application).

You should then go up to the third floor and find a marker on the right side of the elevator. Then you should show the place and talk a little about the floor. (eg "The third floor simply gives access to more DEI professors' offices") (This information will be available in the application buttons).

You should then go up to the fourth floor and find a marker on the right side of the elevator. Then you should show the place and talk a little about the floor. (ex: "On this floor we have access to the rest of the department including the other towers, the amphitheaters and the department bar. Also here we have the laboratories of the CISUC team called SSE, Software and Systems Engineering") (This information will be available in the buttons of the application).

Then go up to the fifth floor and find a marker on the right side of the elevator. Then you should show the place and talk a little about the floor. (eg "The fifth floor has 6 CISUC laboratories where the CMS, Cognitive and Media Systems and ECOS, Evolutionary and Complex Systems teams work".) (This information will be available in the application buttons).

Then go up to the sixth floor and find a marker on the right side of the elevator. Then you should show the place and talk a little about the floor. (ex: "Here we have seven more laboratories where the LCT, Communications and Telematis and IS, Information Systems teams work. On this floor we will take a better look at some of the laboratories and projects that are being worked on there.") (This information will be available in the application buttons).

After entering the floor, the presenter should go through the corridor of the laboratories and look for the first door on the left where the G6.1 laboratory is and next to it will be a marker. You should then talk about the room and the project being developed there. (example: "In this laboratory, a project called "5G-Components and Services for 5G Networks" is being developed, which intends to integrate several new products within the future 5G network that will gradually begin to be introduced in Portugal.) (This information will be available on the application buttons).

You should then proceed along the corridor to the G6.2 laboratory where you will find a marker next to it. You should then talk about the room and the project being developed there. (e.g. "In this lab, a project called "Power" is being developed that intends to create several innovative products and services aligned with many transformative technologies such as 5G networks, artificial intelligence and cloud computing.") (This information will be available on the application buttons).

When you're done, you can turn your back to find yourself facing the G6.6 laboratory where you'll find a marker next to it. You should then talk about the room and the project being developed there. (e.g. "In this lab, a project called "ATENA" is being developed with the intention of creating advanced tools for the analysis and mitigation of critical components and their dependencies on critical infrastructures. ") (This information will be available in the application buttons).

Then you must go back down the corridor to reach the G6.7 laboratory where you will find a marker next to it. You should then talk about the room and the project being developed there. (e.g. "A project called "Poseidon" is being developed in this laboratory, which intends to create a platform that safeguards personal and private information, and also to support organizations in the control and processing of information. ") (This information will be available on the application buttons).

With this lab presented, the presenter will be able to conclude his presentation.

Post presentation talk

“Before we go any further, I wanted to ask if you were able to use the application and all its features?"
"Was there anything you did not understand how it works?”
“Before asking your opinion about the app, I wanted you to perform several small tasks in the app.”
"Can you find out using the application which CISUC group works on the 4th floor?" (SSE)
"Can you see what project is being carried out in room G6.2?" (Power)
“What did you think of the application in general? Anything you want to comment on?”

Final questionnaire

(These question are answered on a scale of 1 strongly disagree, to 5 strongly agree)
-I think that I would like to use this system frequently.
-I found the system unnecessarily complex.
-I thought the system was easy to use.
-I think that I would need the support of a technical person to be able to use this system.
-I found the various functions in this system were well integrated.
-I thought there was too much inconsistency in this system.
-I would imagine that most people would learn to use this system very quickly.
-I found the system very cumbersome to use.
-I felt very confident using the system.
-I needed to learn a lot of things before I could get going with this system.

For Visitors only:
-I felt this system uses Augmented reality well.
-I think the way the information panels were displayed was helpful and informative.
-I found the system added an interesting interactivity between the presenter and the clients.

For Presenter only:
-I thought that the Augmented reality assisted in performing the presentation.
-I felt the information panels helped in performing the presentation.

(These are open answer questions)
-Were there any bugs or problems to report?
-Do you think there are any missing functionalities in the application?
-Do you have any recommendations for this application?

Results

The results of the questionnaire were gathered and an average of the results was made. They are as follows:

I think that I would like to use this system frequently. **Avg: 3.6**
I found the system unnecessarily complex. **Avg: 1.3**
I thought the system was easy to use. **Avg: 4.6**
I think that I would need the support of a technical person to be able to use this system. **Avg: 1.6**
I found the various functions in this system were well integrated. **Avg: 4**
I thought there was too much inconsistency in this system. **Avg: 2.3**
I would imagine that most people would learn to use this system very quickly. **Avg: 4.3**
I found the system very cumbersome to use. **Avg: 1.3**
I felt very confident using the system. **Avg: 4.6**
I needed to learn a lot of things before I could get going with this system. **Avg: 1.3**

For Visitors only:
I felt this system uses Augmented reality well. **Avg: 4**
I think the way the information panels were displayed was helpful and informative.
Avg: 4
I found the system added an interesting interactivity between the presenter and the clients.
Avg: 3

For Presenter only:
I thought that the Augmented reality assisted in performing the presentation. **Avg: 4**
I felt the information panels helped in performing the presentation. **Avg: 5**

With the open ended question and with the post presentation talk there were several topics that were brought up. They are listed bellow:

-Presenter would disconnect when changing internet access point
-Light affected the detection of the markers
-Should be able to resize chat panel and other aspects of the UI
-In mobile you should make the chat panel disappear (need more space)
-Should be able to zoom in the pictures
-Increase text size
-The Presenters webcam panel should be highlighted
-The scroll bar in panel should be different colour than background
-Improve design (change colour pallet)
-Add new types of panels for interactivity (like polls)

Conclusion

The results of this tests showed us that for the most part the users found that the application is intuitive and easy to use. The application follows what is usually expected out of a videoconferencing application and adds the MR elements on top without it becoming confusing. The users also found that the way to add the MR elements is well done and that it adds to the presentation making it more interactive and informative. At the same time the users had a few issues with the application. Most found that the design of the application was not very appealing and that it should be improved. They also thought of a few changes that would improve the application such as adding the ability to zoom in pictures, to hide the chat window, the ability to resize the UI among others. All in all the users seemed to enjoy using the application and were happy to comment on how it could be improved.