

TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Department of Computer Systems

Damilare Olatunbosun Folorunso 184636IASM

Mobile Augmented Reality Application For Control Of Multi-Tank System

Master's Thesis

Supervisor: Aleksei Tepljakov

Ph.D.

Researcher

TALLINN 2020

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutisüsteemide Instituut

Damilare Olatunbosun Folorunso 184636IASM

**Mobiilirakendus mitme paagi süsteemi
laboratoorse mudeli juhtimiseks
täiendatud reaalsuses**

MSc Lõputöö

Juhendaja: Aleksei Tepljakov

Ph.D.

Researcher

TALLINN 2020

Declaration of Originality

Declaration: I hereby declare that this thesis, my original investigation and achievement, submitted for the Master's degree at Tallinn University of Technology, has not been submitted for any degree or examination.

Deklareerin, et käesolev diplomitöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Tehnikaülikooli magistrikraadi taotlemiseks ja selle alusel ei ole varem taotletud akadeemilist kraadi.

Damilare Olatunbosun Folorunso

Date: May 6, 2020

Signature:

Abstract

Mixed Reality has been in existence since the 1800's [1] and has through the years seen a lot of development and advancement in its technology. The advancements in research and technology has helped transform augmented reality (Another form of mixed reality) to what is it today. Thanks to these advancements, the trend has evolved from the big dedicated augmented reality head-mounted display to portable smart devices such as our smartphones , tablets and portable smart glasses. Passive research and developments in the design of smart-phones, tablets and development engines such as Unity 3D, Unreal engine, Vuforia studio, Vuforia engine, ARCore etc, has made it more easier to develop AR applications for portable smart devices to produce better AR experiences for the end-users.

This master thesis is written at the Tallinn Technical University, Departments of Computer Systems. In this thesis, an Android OS based augmented reality application was developed to control a multi-tank system using Vuforia engine for image recognition, Unity 3D development environment , Android studio for APK deployment to an android device and C# scripting language for the back-end design. Using a marker based technology, a virtual object such as a meter gauge with setmarks is overlaid on the real multitank system for the control of the liquid level in the real multi-tank system. The end-user uses a slider on the hosting device screen to move a game-object to control the desired set-point on the virtual gauge. The value of the gameobject is taken as an input and sent as an instruction to simulink to control the liquid level in the real multi-tank system, with the help of PID controllers and a UDP connection to the hosting device (An Android smart-phone) we are able to exchange data between the two communicating device.

This thesis is written in english language and contains 64 pages of texts, 8 sections, 14 figures and 3 tables and 1 appendix.

Nomenclature

AR	Augmented reality.
VR	Virtual Reality.
PID	Propotional Integral Derivative.
UDP	User Datagram Protocol.
3D	Three-dimensional pictures.
2D	Two-dimensional pictures.
MR	Mixed Reality.
UE	Unreal Engine.
Prefab	Creation of a grouped gameobjects as a reusable asset
UI	User Interface.
VO	Virtual Object.
HMD	Head mounted Display
RAM	Random Access Memeory
APK	Andriod Application Package
API	Application programming Interface.
IAR	Industrial application.
HMD	Head Mounted Display / Gear.

Contents

1	Introduction	10
1.1	Background	13
1.2	Problem Statement	13
1.3	Methodology	14
2	Outline Of Thesis	14
3	Augmented Reality Concept	15
3.1	Classification Of Augmented Reality Systems	16
3.2	Augmented Reality Hardware	18
3.2.1	Displays	18
3.2.2	Input Devices	20
3.2.3	Tracking or Recognition devices	21
3.2.4	Hosting Devices	21
3.3	Conclusion	21
4	Literature Review	22
5	State Of The Art	25
5.1	Web AR	25
5.2	Open AR Cloud	27
5.3	Vuforia Studio	27
6	System Model <i>Setup</i>	28
6.1	System Model	28
6.2	Smart-Device	29
6.3	Camera	29
6.4	Review of Smart-device Or Hosting System	30
6.5	Conclusion	32
7	Application Development	33
7.1	Vuforia Studio	34
7.2	ARCore	34
7.3	Unity 3D	35
7.4	Unreal Engine	35
7.5	Conclusion	37
7.6	AR Application Development	37
7.7	Target Image Capturing	38

7.8	Virtual Object Integration	40
7.9	Coupling Of All Gameobjects and Assets	44
7.10	Connecting The Multitank System To Unity	45
7.11	Visualizing Data Between Communicating System	46
7.12	Functionality And Use-Case Overview	46
7.13	Packets Exchange Security	47
8	Conclusion	48
A	Appendix 1. C# script	55

List of Figures

1	Mixed Reality Over The Years	12
2	Mixed Reality Concept	16
3	AR display types	20
4	System Model	29
5	Simple Mobile Camera.	38
6	AR Camera	39
7	Vuforia Engine Image Rating.	39
8	Target Image In Unity 3D	40
9	Unity 3D VO Without Material	42
10	Unity 3D V.O With Material	42
11	Overlaid Virtual Object	43
12	APK Deployment	43
13	Coupling Of Game-object	44
14	PID-Controller Model	45

List of Tables

1	AR inventions.	11
2	MI Mix 2S	33
3	Dev-Engines	36

1. Introduction

Augmented reality can be defined as a real-time view of a real-world environment through a technological device, which is supplemented by computer-generated imagery such as images, text, audio, animation 3D/2D graphics[2], in a simple term it is the notion of the real or physical environment in which elements are enhanced by computer-generated elements. This area of technology is no longer science fiction, there are now various portable off the shelf smart devices such as our mobile phones, tablets designed with the capability or enhancement to run AR applications[3]. The use of augmented reality for industrial applications has been in existence decades ago, it was first applied in the industrial sector in the 1900s by a Boeing researcher Tom Caudell [4][5] and has through the years been embraced by other industrial sectors. This area of technology has evolved from the bulkiness of head-mounted device/displays and backpacks to our portable smart mobile [6] devices such as Google Translate, Pokemon Go, IKEA Place, Google maps etc. Augmented reality combines the real world we currently live in with virtual elements in other words giving another perspective to the way we view and interact with our real environment. The computer-generated augmentations are projected on top reality in a way that is supposed to help us with our variant tasks and help us with being more efficient in our day to day activities.

The history of augmented reality or Mixed Reality spans beyond decades. It dates back to the 1800s. Antecedent to the technology we have today, MR be it AR or VR goes way back possibly as far as humans had been using sounds and arts or images which is what we know today as Analog AR [1]. However, this practice has through the years evolved to the current state of the art, all thanks to further research and constant advancements in the technology and concept behind AR [7]. Through the past years, there has been constant development in this area of technology, the advancement in technology help the trends in the 1800s evolved from big head-mounted displays with backpacks (HMD) to smaller augmented reality gears which are a little more portable compared to the gears used in the 1800s. improvements in this area of technology continued into the 1900s however the HMD were still trending but the main development was the introduction of interactive concept for more user experience rather than the previous, which was simply displaying of a virtual element to the user, an example of this was the first flight simulator in 1928.[1]. The trends in the 2000s brought in the possibilities of much more portable and compact gears

with more personalized augmented reality experience to the general public. This area of technology contributed to the boom of the gaming industry as more Mixed reality applications were much more accessible to the public due to the improvements in mobile and smart devices used during the decade.

Table 1. AR inventions timeline.

Year	Inventions / Achievements	Inventor
1968	First head mounted display	Ivan Sutherland with Bob Sproull
1974	Videoplace	Myron Krueger
1990	AR application to aid electronical wiring in an airplane	Tom Caudell
1992	First real operational augmented reality system(Virtual Fixtures)	Louis Rosenberg
1994	First theater production to use augmented reality (Dancing in Cyberspace)	Julie Martin
1998	1st and Ten line computer system	Sportsvision
1999	Hybrid synthetic vision system	NASA
2000	ARToolKit(Software)	Hirokazu Kato(Nara Institute of Science and Technology)
2001	Skycam	Garrett Brown
2009	Print Media	
2013	MARTA app	Volkswagen
2014	Google Glass	Google
2015	AR market reaches \$700 million investment	–
2016	Hololens	Microsoft.

Augmnted reality is no more science fiction. This technology is now incorporated in most of our portable smart devices such as mobile phones and tablets, however, the future goals of this technology are already being implemented a across variant areas of the industries, the prospects for augmented reality is that at a certain stage in research and development contribution, there would be hands-free mobile applications which might help solve some of the issues that hamper the development of the technology[3]. Issues such as social acceptance and on-time use case. The improvements of the technology has put it the concept of augmented reality in

a position where various industries now embrace the technology to help solve or simplify problems in areas such as the medical industries where augmented reality could help overlay X-ray images on patients before or during surgical procedures. Mechanical industries have also adopted this technology as industrial workers are not trained using AR gears the help navigate tasks by overlaying virtual parts to help understand complex repair works, most industries do utilize the head-mounted displays as it allows the users use their hands and some other industries utilize the smart devices as it gives a sense of touch and brings the virtual object closer to the users facilitating more interactivity. The notion of hands-free augmented reality gear or application would help boost the advancement of the technology but its development would be challenging as the most head-mounted displays and smart devices development have not been perfected to the stage or the required level[8][9].

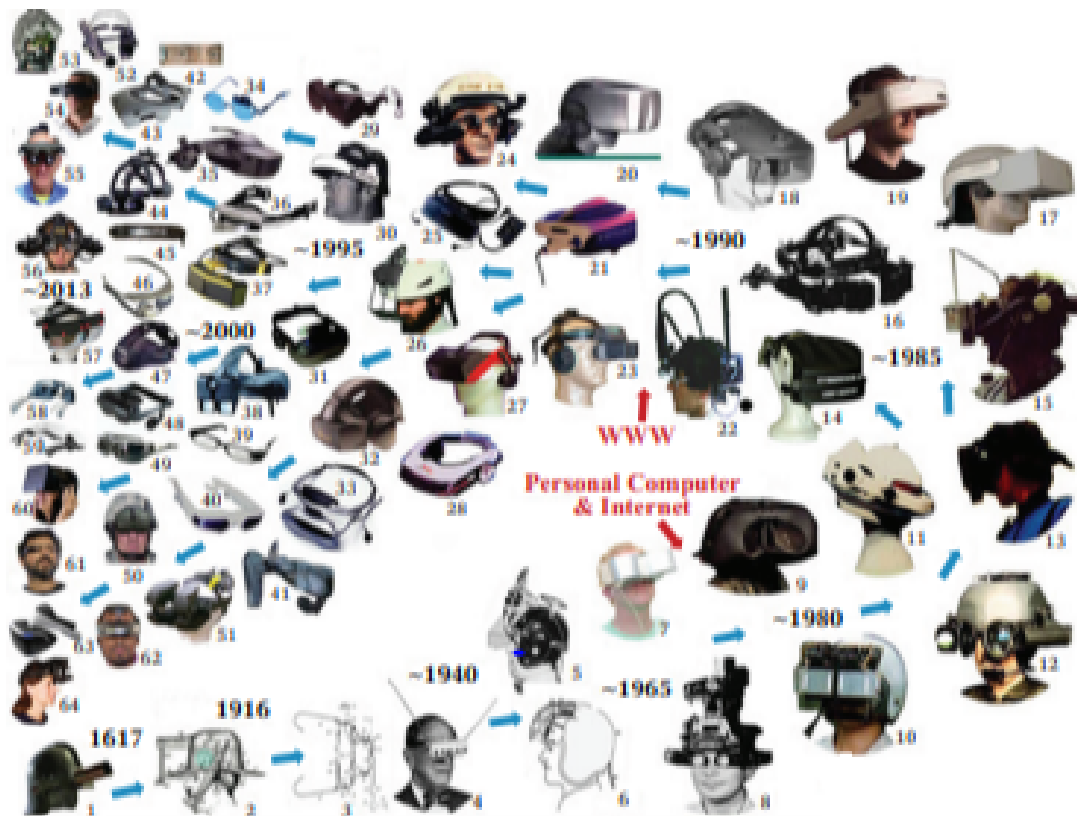


Figure 1. Mixed Reality Over The Years

This immediate thesis is devoted to the study of design and development of augmented reality applications for possible industrial application. Marked and Markerless augmented reality application method will as well be considered[6][10], ways in which augmented reality could help control systems in the laboratory would also be reviewed.

1.1. Background

The use and importance of this technology have been growing at a proliferating rate and being applied to many other fields as it is rich with a lot of yet to be discovered potentials. Augmented reality applications allow it's user to view the real environment from a designated AR-enabled camera and display designed virtual object overlaid on the real-world object, giving the user a sense of being able to interact with the real world. These methodology has already or being incorporated in some fields mentioned below:

1. Medical Training, Public Safety.
2. Retails and Logistics.
3. Military.
4. Repair and Maintenance.
5. Design and Modeling.
6. Media, Gaming.
7. Industry 4.0

The flexibility in the developments of an AR application can allow us develop a simple but yet efficient mobile-friendly application for end-users and various industrial use cases, as an example one can create a virtual representation of an industrial unit model for the purpose of explaining a task to employees or creating a virtual representation of an end product to be presented to customers, in other words, a digital twin of an object in real life.

1.2. Problem Statement

This thesis is directed towards the possibilities of creating an Andriod augmented reality application for the control a multitank system in the XR lab of Tallinn University Of Technology. The aim of the thesis is to design and develop a mobile AR application with its unique virtual objects to control of a multi-tank system.

For this to be possible the problems below should be tackled:

1. Selecting the best and capable hosting smart device.
2. Selecting the best development engine for designing.
3. Unification of virtual objects and the target image.
4. Integration of the virtual object into the application.
5. Deploying the application unto the hosting device.
6. Establishing a remote connection to the real world object for remote control.

The end result would be an augmented reality application with UI elements which uses a marker-based technology to identify the target image (Multi-tank system) and overlay virtual objects onto the real-world object, sends the users input to MATLAB Simulink for tuning of the real multi-tank system.

1.3. Methodology

This thesis is based on 2 methods namely practical approach and criteria approach. The practical approach meaning much more attention was paid to the development and designing of a presentable application and the criteria approach meaning certain choices were made based on research and based on these choices we selected the best approach which best solves our problems.

2. Outline Of Thesis

The thesis is ordered in a way to provide detailed clarifications of development stages for logical presentation.

The first chapter describes the route of the thesis, background, problem statement and methodology used for the thesis.

The second chapter holds our thesis outline and describes the structure of our thesis.

The third chapter describes the concept of AR, classifications, the hardware we further discuss the different types of displays, input devices, image recognition and different types of hosting device also known as our smart-phones in our thesis.

The fourth chapter is devoted to the review of past literature and researches in making the AR better.

The fifth chapter describes the State of the art, where we look at some of the current advancements in the technology of AR.

The sixth chapter describes our system model and how we structure our AR application, we further reviewed our smart device, the quality required for our AR camera.

The seventh chapter describes the steps taken while developing our AR application, we further reviewed the different frameworks and development engine we could use and reviewed which best suit our AR application. The chapter also describes the process of our target image capturing, virtual object integration, deployment of our APK application to our hosting device and made some test to ensure the application works as it's supposed to.

The last chapter was devoted to our conclusion where we once again poured over the initial questions we had at the beginning of the thesis and based on our sequential reviews we were able to achieve our goal for this thesis.

3. Augmented Reality Concept

Most users or the public still mix or confuse VR and AR. This technology however in a way complement each other as they both produce MR experiences for their users, but it is worth mentioning that there are differences between these two types of technology. Figure 2.2 shows the various forms of reality, the notion behind AR, VR and the real world is known as MR. The difference between these forms is that the real environment is particularly what we see and what we can feel, AR relies on

the real environment as this technology is used to complement a real live objects, VR is variant as it is simply a virtual world where what is seen in VR is virtually unreal.



Figure 2. Mixed Reality Concept

3.1. Classification Of Augmented Reality Systems

AR applications are developed to solve variant problems, produce certain types of mixed reality experience or tell variant stories and produce variant experiences, we can catalogue augmented reality based on the type of it's information representation, the various forms are

1. Audio: These are dedicated systems that produce sounds for the enhancement of it's user's experience. They are mostly used to aid navigation[11], an example is the trending use of audio VR gears to narrate stories to the users while touring the museum an example is "Guided by Voices" this application is a simple wearable computer that uses radiofrequency to provide audio geographical information to it's user[12].
2. Visual: These are dedicated augmented reality enabled gears, such as Hololens and our smartphones. The augmented reality gears include a camera for visual or image streaming which can also be used to identify a target image and once the target is recognized the virtual object is overlaid in the real world for the user to see.
3. Audio-visual: This is the combination of the other two types of classifications, it combines both the visual and audio senses to give a rich and complete augmented reality experience to the end-user.

As this technology depends on information and data from the end-user and it's environment, these systems would need some set of sensors for gathering data and

processing them accordingly, data such as electromagnetic waves for navigation, heat sensors, sound, acceleration and so many more based on the function of the function the application is to perform.

To further classify these concept, it is possible to also categorize based on their functionality or purpose as same sensors could possibly process and trnasmit the same type of data. Based on the type of sensors we can further classify as:

1. Audio systems for processing users voice commands and voice comands and messages.
2. Visual systems for processing visual effects and processing of i mages from the camera.
3. Positioning systems for signaling locations. These systems can be used as accelerometer or as a compass for navigating.

We can furthermore classify augmented reality based on the degree of interaction it has with its user. In some cases, the users do play most of the roles that is the user dictate the behaviour of the system and in the other case, the user depends on the user to give it instructions on how the virtual objects are displayed, where it's displayed, and how the systems behave/responds to commands. The classification can then be either:

1. Interactive : These type of systems do engage the user thus giving the user more AR experience. The type of applications or systems take input from the user[13], process them give back an output which in turn makes the user feel more involved in the workflow of the system or applications.
2. Autonomous: These type of systems or application does not require any input from the user. This type of systems are mainly to provide information to the user[14]. Example is the Layar application, this application allows the user point their smart device camera at a particular target object and then a pre-programmed vitual object or information is overlaid and fed back to the users screen.

The degree of interactivity can be as well be further be described based on the functionality the application offers the user, there are applications designed to allow

their user change their virtual objects the user wants to interact with[13]. Google AR[15] core allows augmented reality developers integrate a dedicated virtual object database with their applications by doing this, their users can choose various virtual objects they did like to interact with, this type of augmented reality experience can be used to replace the real object when it can not be used, an example is a specialized application for a specific medical procedure[16]. There are as well other systems that do not require or allow the user to change the virtual environment or virtual object, but does give the user the ability to tweak, move, resize, rotate the environment or the virtual objects. This type of application or systems only allow the user manipulate the virtual object at the display layer i.e image rendering. Such application are commonly used in advertising, architectural exploration[17].

Further classification of systems based on it's mobility features.

1. Mobile: These type of systems allows the free movement of the devices used to produce the augmented reality experience. The free movement could as well be based on the type of functionality of the systems.
2. Stationary: These type of systems are variant from the portable ones. The type of systems do not offer or does not allows free movement of the devices used to provide the augmented reality experience an example is Meta 2 HMD.

3.2. Augmented Reality Hardware

Various gears can be used or are used in providing AR experiences and integrated for augmented reality experience, however the main devices are mostly a screen for display, speakers and microphone for audio processing, camera for recognition and visual streaming to mention a few[18]. In this chapter we would be discussing some of the gears used in development of augmented reality applications.

3.2.1. Displays

There are generally three main types of displays options for augmented reality experience [19]. The type of application functions determines the type of displays to be used, the main types of displays are generally handheld display, spatial display

otherwise known as projectors and head mounted displays. Out of the three, the most common display is the hand held display as it is the cheapest as it's mostly readily available and the most portable and currently in trend. They are mostly small screens display the users can hold with either their both hands or in a single hand, large enough to let the user touch and see the virtual object overlaid on the screen using the video transparent technique.

The Head mounted display is a type of display designed specifically for augmented reality experience. The Head mounted display like the name implies is a gear consisting of straps, frontal glass or screen for image overlay, and some hearing aid all put together and worn on the users head. The real world is displayed to the user through the glass in front of the users view, this screen as well is used to display the virtual object for the user to see. The Head mounted display also contains a camera in front of the gear the helps track and identify images in the real world [20].

Spatial display works by using projectors, optical elements such as holograms and other technologies to provide graphic information to the physical object thus does not require the user to wear a HMD or hold any hand held display [21]. The figure 3[22] below shows the comparison of the different types of displays as well as the cons and their pros, the type of technology used in this displays [23].

Type	HMD		Handheld	Spatial		
Technology	Video transparent	Optical transparent	Video transparent	Video transparent	Optical transparent	Direct supplement
Advantages	Visual control, synchronization of the virtual and real environments, waiting video processing from cameras	Half silver mirror technology, the natural perception of the real environment	Portable, powerful, combination of many sensors	Price, can be adapted to the standard equipment	The most natural perception of real objects	Display directly on the surface of a physical object
Disadvantages	It is necessary to fix the device on the head, unnatural perception of real world	The time delay, jitter of virtual elements	A small display, weight (for tablets)	It does not support mobile systems	It does not support mobile systems	Do not depend on the user (everyone can see the same thing)

Figure 3. AR display comparison

3.2.2. Input Devices

Various types of devices can be used as an input device. There are applications or systems that could or are designed to work with input devices such as wireless bracelet, dedicated augmented reality gloves or another peripheral device attached or connected to the system such as the hololens clicker. In the applications designed for smartphones, the screen itself is the input device as it allows the user to touch the screen and move the virtual object about, however the type of input devices depends on the type of the systems, application as well as the function the application is meant to perform.

3.2.3. Tracking or Recognition devices

For the tracking of objects in the real world, the system or the application would require an optical capable device such as a camera, heat sensors, GPS, accelerometer to mention a few. As these systems vary in their accuracy and processing power, the type of the tracking device used in an application or systems would as well depend on the type of application function. The devices alone do not track and perform the image recognition, this is done by algorithms or image recognition engines.

3.2.4. Hosting Devices

For the best-augmented reality experience, the hosting device would be required to have a good amount of processing power, enough RAM, long-lasting battery capability, Video memory for processing videos and images quality, good audio quality, but thanks to constant research and development of the new generation smart devices we now have much more capable systems designed to handle complex augmented reality applications.

3.3. Conclusion

In this section, we analysed the basic attributes of augmented reality application as well as other possible required devices that could be used to provide an augmented reality experience for the users. The theoretical information helps define the next attribute of our applications.

1. Interactivity.
2. Visual type.
3. Smart device / hosting system.
4. Visual type
5. Display type.

The points mentioned above are taken into consideration when making choices while developing our AR application for our thesis.

4. Literature Review

There are numerous research papers dedicated to improving the technology behind augmented reality and the ways developers can get the best of the technology. Most of this theoretical and practical contributions indicate that augmented reality can and will in a way contribute to the advancement of any industry the technology is applied to [24],[25] . The research and development support community has contributed a lot of new ideas to the development of AR, the year 2014 was a year that saw a huge financial investment in this area of technology. The contributions thus lead to an increase in interest and demand of augmented reality applications in industrial companies, however social media platforms, video games industry, and marketing industries contributed immensely to the distribution of augmented reality in local fields such as fashion, food, customer service, and gaming. Nevertheless, as mentioned earlier, the leading sectors remained the industrial ones like manufacturing, automotive as more researches are been conducted for the improvement of AR technology. This section holds the review of exciting developments and research in mobile AR applications.

The development in the area of technology is expanding beyond the current trend in the development of AR applications and it's use cases, however, some problems still need to be solved. The problems could be classified into 2 namely:

- Technical challenges.
- Non-technical challenges.

The technology used in developing AR application is evolving, thanks to various researches and contributions of various researchers and developers, there are better resources and materials available for the public to utilize and explore. Researches have contributed to the development and public awareness of the concepts of mixed reality, exposure to the advantages AR has when blended with processes and ways to dynamically develop AR applications for variant use case scenarios. The contributions

and researches can be broken down into three domains based on the achievements and outcome solutions.

- Technical contributions and researches into hardwares and software used or that can be used for the development of the technology.
- Research and Solutions to end user related issues.
- Research into the practical use case scenarios.

Research paper by the authors [26] gives an introduction to the concept of augmented reality and describes approaches to find solutions the users of AR applications might face when using the applications as it still unclear how long and if this technology poses any strain or stress on the end-users of this applications. The authors proposed a method based on analysis of the Heart Rate Variability to objectively measure stress and strain the application has on the end-user while performing various task. The authors examined various result from the research conducted in a warehouse using people without package picking and sorting experience, the twelve test volunteers were given an AR optimized device to help them identify packages and at the beginning of the test, the users all showed signs of stress which brings about the theory that the end-users experience this strain as they are trying to comprehend and how the AR applications work, which is a common experience most end-users face at the initial stage of using AR applications. The authors propose a seamless approach at the development of AR applications by ensuring the end-users does not have to put in much effort to work with the application which gives the ideal that AR applications should be automated and much as possible so it can encourage more user interactions and reduce the time it takes for the user to get accustomed to the application.

Most AR applications are mostly dedicated to use case scenarios such as Maintenance and remote assistance, most of these types of AR applications has limited user interactivity which limits the end-users to mostly features such as touch hear and viewing 3D images. The authors of [27] developed an application that enables maintenance workers to perform the same task with an extra AR features such as speech and voice command recognition. The authors developed an AR application connected to a plant management system through wireless connections, the developed application consisting of 3D Virtual Reality Markup language, speech recognition and speech-synthesis engines which allows a maintenance worker communicate and

command 3D virtual objects in the AR scene facilitating navigation. These methods give the end-users more functionality as he / she can interact with 3D objects while performing other activities with their hands.

There are two ways AR applications work, the common method is through the use of target image, when an AR camera identifies a tagged image it displays a pre-designed virtual image for the user to see, the author of [28] proposed a solution for AR based repair guide using the Marker-less AR system. The concept is different from the first method mentioned earlier as the marker-less technology does not require any tagged image for the AR camera to identify. The authors proposed a solution to help solve hardware and software issues when using such systems, the authors used a markerless tracking algorithm based on 2D point features and uses a CAD model to establish 2D-3D correspondences which is used to calculate the position of the camera pose relative to the real object in the real world. The end result is a prototypical AR application which utilizes markerless tracking with a monocular camera setup as well as a light weight mobile hardware platform for visualization with the drawback being from error accumulation during the tracking stage due to lighting conditions around the device.

The research paper by the authors of [29], investigate a crucial part of the industrial sector known as intralogistical planning in the automotive industry. The need to innovate processes in the manufacturing and delivering product to the consumers arises as consumers constantly demand more comfort and curvy processes. The authors investigated reasons there are little or no AR applications in intralogistics planning and based on their finds proposed essential points in developing AR application for this part of automotive industry with a prototypical solution and finally discussed the advantages the integration of AR technology can provide the industry. The authors concentrated on the development of AR applications for portable smart devices such as our smartphones and tablets as they are universally without being restricted to geographical locations. The authors developed an AR application that supports logistical planning in the final assembly in an automotive industry as intralogistics planning process consists of many sub tasks by including the process of visual layout planning and material flow planning. The advantages of the application is that the introduction of AR to the process flow helps the companies plan better as it gives the ability to visualize their ideas before they are implemented in the real world, one more advantage of introducing AR technology is that the data derived from the application can be streamlined into any part of the other systems to help make

informed decisions and predict future problems.

The authors of [30] analyzed the integration of different hardwares and softwares used in IAR for the development of AR applications for shipbuilding industries. The research is part of the Shipyard 4.0 project which signifies the application of the industry 4.0 principles to optimize shipbuilding processes with a user-friendly interface and information on relevant processes to shipyard workers. The authors presented an IAR architecture which can be applied to other sectors of industry 4.0 use cases to reduce the latency responses for a smoother and more realistic AR experience to the end-users. As a contribution to the development of IAR the authors proposed an architecture for designing of AR applications that would work in an environment where there are electromagnetic interruptions. The authors developed an IAR application which can be used in scenarios such as Plan information, Quality control, virtual assistance, Asset location / finder, virtual Installations guides and Augmented collaboration and reporting to mention a few. The research indicates that there are many benefits IAR can offer industry 4.0 as it can make processes a lot more efficient, reduce time effort in optimizing processes.

5. State Of The Art

5.1. Web AR

Mobile Augmented reality is the trending norm in the field of Mixed Reality as developers are getting better at developing AR applications for cross platform functionality. The common methods of mobile AR are either hardware based mobile AR or APP based mobile AR. Hardware based mobile AR is costly, time consuming and most of all difficult to develop due to challenges such as flexibility as hardwares are configured in different ways, so making them compatible would require more effort which contributes to the time invested in developing an AR application dedicated to the hardware components. To solve these problems the concept of a Web based AR application will definitely help minimize the challenges that hamper the developments of AR. There are researches and prototypical applications that help demonstrate this concept and how it could be done efficiently. The authors of [31] give an elaborated view of the Web AR approach, there are researches conducted such

by authors of [32][33] and many more have been contributed towards the trend of decentralized AR experiences. Most mobile AR application lack cross platform functionality and to accomplish this developers mostly have to redesign the application for it to accomodate more platforms which increases the amount of time, cost and sometimes more problems when making this features. Thanks to the advancements in the designs and funtionalities of our smart phone and tablets, problems that might amper the development of Web AR are reduced as there are better coonectivity such as 5G and mordern smart devices are equipped with some of the best sensors, larger displays and cameras for better visaual capablities and most of all more powerful computing power as compared to what we had before. The authors of [31] deccribes the challenges developing AR applications with Web AR as solutions to the current methods of development. challenges such as limited computing power compared to the requirement of powerful computing, coonectivity lags compared to Real-time performance, limited power compared to power consumption, the authors describes the possible solutions to this challenges and ways to help reduce the impact these topics might have on quality of Web based AR applications.

As this conscepts of AR is now attracting the attentions of both researchers and developers likewise, ways to effciently implement this concept can be categorized as below:

1. Self- contained methods which is the typical method for Mobile AR development which allows smooth computation process, however and the emebded components within the hosting devices are with limited capabilities, the form suffers from lags, connectivity problems and tradeoffs.
2. Computation out-sourcing take advantage of remote storage and some com-pautational requirements therby increasing the efficiency of the hosting device.

As mentioned ealier, Web AR is getting more attentions and as more reserches are conducted and develeopers exploring this concept, it is certain that in the coming years Mobile AR / AR technology would difenately get better than the current state of the art.

5.2. Open AR Cloud

This concept still at its developmental stage hopes to bring more sophisticated solutions to the challenges hampering the development of AR. Open AR Cloud is a persistent 3D digital copy of the real world. It is a collection of billions of machine-readable datasets, point clouds, and descriptors, aligned with real-world coordinates, a living, shared, “soft copy” of the world created by scanning physical features around us in which persistent augmented reality experiences reside[34]. Most development AR applications are always individual or standalone applications which tends to serve a single purpose. Open AR Cloud provides the opportunity and platform for different AR applications to access Data and resources needed for crossplatform functionality and use cases. This concept or development will help solve the technical challenges such as memory usage, connectivity, single use cases of application and reduce time spent on developing AR applications. The three major components of AR clouds are:

1. Scalable and shareable machine readable representation of the world aligned with real world coordinates.
2. The ability to instantly align the world’s soft-copy with the world itself from anywhere and on multi devices.
3. The ability to place virtual content in the world’s soft-copy and interact with it in realtime, on-device and remotely.

5.3. Vuforia Studio

IoT is a major component required to build or develop good scalable AR application. Industry 4.0 relies on interoperable machines for better and faster manufacturing processes. Vuforia recently released a new development engine for the designing and development of AR applications for industrial use cases. The new development engine comes with a more resource development kit and much more user friendly user interface. Vuforia has through the years work to help solve problems such as image recognition by providing packages which could be integrated with other development environment such as Unity and Unreal engine. Vuforia Studio provides industrial

companies with the tools to quickly create and widely deliver AR content that transforms operations, facilitates knowledge transfer and improves profitability[35].

Features that makes Vuforia Studio better than other development environment are:

1. Drag and Drop authoring saving time designing virtual objects.
2. Online and Offline user AR experience.
3. Advanced spatial and model tracking.
4. Inbuilt cloud hosting and On-prem.
5. Assisted Reality.
6. Native IoT integration.
7. Global Scalability.

6. System Model *Setup*

This section holds the description of the required components, features, and attributes of the application discussed in this thesis. We would also discuss the process of integrating the hardware components and the softwares used for the development of the application. To better describe the process we would start with the description of both the hardware components and the third-party applications, frameworks and softwares we would be using. We would also be comparing various smart devices to use for our application, frameworks as well as intergration and implementation methods.

6.1. System Model

To develop the andriod application for this thesis, we would be following the pattern below in Figure 4 . The APK is developed unity Unity 3D developepment engine and Vuforia SDK for the image processing and image recongnition as we would be using a Marker based technology for the identification of our multitank in the real world.

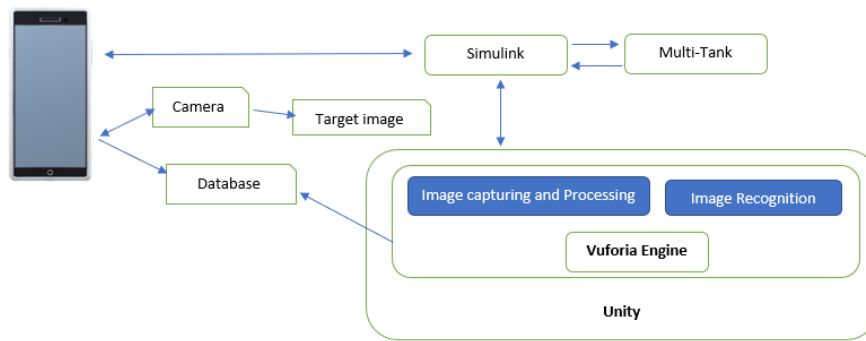


Figure 4. System Model

6.2. Smart-Device

The smart device consists of some other systems or component such as displays, integrated camera, integrated battery, memory, network connectivity, as well as speakers to mention a few. The designed application is developed and virtual object are developed on a personal computer and then uploaded onto the smart device as an APK file for installation on the smart device. There are various types of smart device that can be used , this types of smart devices range from different types of high-end mobile phones to some of the enhanced handheld tablets. There are now variants of smart devices that now have enough processing power, long battery life, better network connectivity and fast responsive instruction processing capability, with this enhanced features we can in a way limit the technical challenges of augmentedreality.

6.3. Camera

For the complete experience of augmented reality, the hosting device in our case has to have a camera either inbuilt camera or external camera. The need for the camera is to make and identify the object in the real world. The hosting device in our case a smart phone or a smart tablet uses an inbuilt mono camera. The mono- camrea is different from the stereo camera as it uses two individual cameras to capture the same image thus simulating the human vision. There are challenges that our augmented reality application faces in-terms of camera. Most smart devices has inbuilt cameras and with not very good quality. The better the cameras on the smart device the more expensive the hosting device cost. We also consider using

external cameras such as the common Logitech HD Webcam which is relatively easy to integrate with any AR plugin or development engine. The challenge with the use of external camera is that it would require the user to always carry the camera around. We thus decide to go ahead with the in-built smart device camera as it is much more compact and portable for the user to carry around without needing to move the camera around or connecting external gears to the hosting device as most smart device has all required gears and parts embedded already.

6.4. Review of Smart-device Or Hosting System

There are many smartphones and tablets out in the market which can now host and run AR applications better than their predecessors. To get the best AR experience on such devices, we would have to select the best smart device which has enough processing power, memory capacity, screen size for better display and quick responsive touch sensors and better battery capacity to mention a few. In this section, we would compare the different types of smart device variant we have in the market. However their smart device with lesser features mentioned above would still be able to provide the same AR experience but with many difficulties such as noticeable lags, faster battery drainage and crashing application to mention a few. In this chapter, we would review some of the new smartphones and tablets that would run AR applications with little noticeable issues.

Apple iPhone The company [36] in the past years has some of the best devices have enhanced features embedded into the device which allows it to perform well when running AR applications and producing better AR experience to the user. The device has a good True-Depth Camera which allows robust face tracking, faster detection of object positions, topology and gestures with high accuracy. The device also has better scene understanding and lighting estimation as it can find and analyze scenes presented to the camera. also, the device had visual-inertial odometry which is fused with the camera sensors for better sensing of object movement. The device also has a 6.1 Liquid Retina HD display which is enough to display various virtual object for the user to see. Lastly, the device has high-performance hardware and rendering optimization[37].

Samsung Galaxy S9 Samsung Electronics [38] has been one of the contributors in the development of AR. The company recently partnered with Google to extend the ARCore platform to more Galaxy devices thereby bringing more AR experiences to their users. The device consists of redefined camera standards with dual-pixel sensors, advanced optics and large screen ratio displays. The device also includes Bixby vision for better understanding of the world through objects, scene and text so it can identify objects and provide new ways of understanding and engaging with the world around its user. The smart device has a 6.2 inch quad HD super AMOLED display for better viewing angles, it has stereo speakers tuned by AKG for better audio playback, the device also comes with an in-built 3500mAh battery. The device also comes with a 6 GB RAM and a 64GB expandable memory, These technical features help the device run AR applications with little noticeable lag as AR applications take a lot of processing power.[39].

Asus Zenfone AR (ZS571KL) This smart device with an Android operating system, is a device built specifically for augmented reality by ASUSTek Computer Inc [40]. The device is the world's first 5.7 inch display ratio with a Tango and Daydream ready phone designed by Google. Tango is an AR technology that changes the way we interact with the world and expands our vision of the world around us as it uses motion tracking sensors, area learning as well as depth perception to better understand the device surrounding and easily identify objects in the real world. The Daydream technology is to give high-quality immersive mixed reality experience without the need to wear a VR headset or AR headset. The smart-device comes with a Tri-camera with a 23mp that allows the device to create a three-dimensional model of its surroundings. Finally as mentioned earlier the smart device is the world's first engineered smart phone to run AR applications as it comes with an optimized S821 CPU and a 8 GB RAM. This technical features engineered into the device makes it one of the best devices to host and run most AR applications[41].

Google Pixel Slate The device produced by Google Inc [42]. runs on a Chrome OS operating system. It has a 12.3 inch display for wider view and better HD display, the device has a 293ppi and a 6 million pixel display. It has an Intel Core i5 processor with a 16GB RAM and 256GB solid state memory space. The device comes with a 48Wh battery which could last up to 12 hours of battery life, the device comes with a 3-axis gyroscope, accelerometer, hall-effect sensor as well as an ambient light sensor. The device has the option of working as a tablet or switching to a mini

laptop. the feature allows or gives the user the option to actually customize or design an AR application as it support the ARCore API for cross platform development[43].

IPad Pro Apple's Ipad [44]has some of the best cameras and processing power, all the models of the new iPad Pro models all support AR as they all have features that helps them host and run AR applications well so as to deliver the best AR experience. iPhone and iPads have been custom-tuned with cameras that are great for shooting in low light and at 60-fps video, which makes it the perfect speed for augmented reality applications and optical image stabilization enables greater visual clarity, so virtual elements can appear more naturally in the users environment . The apple products uses the A13 bionic processing chip as it is also designed for powerful CPU efficiency. The device also comes withe better image signal processors which enables real-time lighting estimation, making virtual objects appear incredibly realistic[45].

6.5. Conclusion

After varioius consideration of different possible hosting device, we decided to select the Xiaomi MI Mix 2S [46] as our target hosting machine. The smart device is equiped withe some of the latest technologies and proved capable of running and producing similar AR Experience after various tests in the lab. The smart device produce good images and was very responsive. The Table below shows the smart device specifications which made it produce a good AR experience [47].

Table 2. Hosting Device

Component	Features
Processor	6GB RAM + 64GB Storage, Qualcomm Snapdragon Upto 2.8 GHz.
Display	5.99 Inch (2160*1080 FHD, 403 PPI, 1500:1 Contrast Ratio).
Operating System	Andriod
Dimensions	150.86mm Tall, 74.9mm Wide, 8.1mm Thick 189 G Weight.
Battery	3300mAh Inbuilt Li Polymer Battery.
Camera	12MP Dual Wide-angle Lens, 4 Axis Optical Image Stablization
Connectivity	Support 2.4G Wifi, 5G Wifi And Wifi Direct. Bluethooth 5.0 USB Type C Port.
Sensors	GPS, Ultrasonic Proximity, Gryroscope, Accelerometer, Ambient Light Sensor, Hall Sensor, Barometer.

7. Application Development

There are various types of framework or game engines that support the desginning and development of both AR and VR applications, development engines such as the famous Unity 3D development engine, Unreal engine, AR Toolkit and the new developemt engine Vuforia Studio which was released in early 2016. The functions and types of the AR applications dictate the type of frame work to use as this engines all have different features, libraries and supporting applications, however all of this framework all support bothe the creations of different type of MR.

For this thesis we would need the option to use existing virtual objects or possibly creating our own virtual objects, import target images and data as well as establishing communications to some 3rd party applications and objects, with these options we would have to review the available development engine that would best help ease the designing, testing, deploying a functional android version of our stand alone application to control the multi-tank system in the real world.

7.1. Vuforia Studio

Developed by PTC, Vuforia studio is a new development engine released early 2016 for creating sophisticated AR applications and experiences. The development engine is primarily dedicated to industrial companies to help them create quick AR applications to promote productivity. The development Engine comes with extra features to help application developers create an application fast, thanks to the Drag and Drop features, advance Spatial and Model tracking, cloud hosting, clean and user friendly interface, support for all deployment platforms, as well as Assisted reality to mention a few, the editor helps reduce the development time and cost as all elements, parts, plugins and engines are all embedded in vuforia Studio[35]. Vuforia studio is expected to be one of the best development tool for AR applications used in manufacturing industry in the coming years as it is specifically designed to work with the Internet of Things service and 3D modeling.

7.2. ARCore

Developed by Google as an alternative to ARKit. The SDK can be used to design AR applications for all Google platform. AR Core in a way helps solve the challenge of a single purpose application as it uses different API to help our smart devices sense their environment, Map their surroundings and interact with the environment. AR Core API support both android and iOS platforms for cross-platform AR experience which make it possible for different people using the same application experience different experience. AR Core works by combining three features essential for good AR experience, the three features below are utilized and integrated with virtual content and the real world as presented to the hosting device input channel which is the camera[48]. Example AR applications Developed using AR Core are Ikea Place,

Pottery Barn 3D Room View to mention a few.

1. Motion Tracking for the sensors in the hosting machine to track it's real time positions.
2. Light estimation allows the hosting device predict the environment lighting condition.
3. environment al Mapping for estimating the size, shape, location of all types surfaces either vertical or horizontal and other types of surfaces in the real world.

7.3. Unity 3D

Developed by Unity Technologies, the editor is one of the most famous engine used by many developers to create either 2D or 3D applications. The editor is one of the fastest to create a Mixed reality application, the editor comes with free and paid asset store for any type of application it's users are trying to create. The Editor works across different platforms such as Windows and iOS and applications developed with Unity can be deployed to any type of hosting device. Unity, in particular, makes application deployment simple as all it works with various third-party applications[49]. Example appliaction developed using Unity are Pokémon Go, Wheel Replacement Training by PROGRAM ACE[50].

7.4. Unreal Engine

Developed in 1998 by Epic Games, it is mostly used for game development as it is capable of producing the best picture quality and better object response. UE is a massive system of tools and editors and several components working together to produce some of the best MR applications. Components such as Sound Engine, Physics Engine, graphics Engine, input and Gameplay framework as well as realtime module are all used in UE. One of the advantage UE has over Unity is that analysis of how the application works can be observed before it's deployed onto the target device so developers can actually correct and fine-tune their applications before

deploying which is not possible in Unity editor [51]. Example of AR application developed using UE are the famous Pubg mobile game and Fortnite.

In the table below we would compare the different types of available development Engine and why Unity development Engine was selected to be the best choice.

Table 3. Development Engines

Dev-Engine	License	Backend	Uses	Platforms	Pros	Cons
UE	Free, royalty fees of 5% for > \$3,000 income after publishing of application.	C++	2D,3D,VR, AR	Consoles, Mobile, PC, Web browsers	Robust and high level graphical displays	Steep Learning curve
Unity	Free if revenue <\$100,000 Pa	JavaScript, C#	Better 2D, 3D,VR, AR	Consoles, Mobile, PC, Web browsers	Easy to learn, User friendly interface, Enormous Asset store, Extra plugins	Limited functionalities for the free version.
Vuforia Studio	Free educational version				Assisted Reality	

7.5. Conclusion

After various thoughtfulness of these frameworks, Unity was selected for the development of the application used in this thesis. Unity was the best choice due to its features such as a large collection of games assets which are available in free and paid versions, mobile-friendly resource utilization, strong programming language C#, portability and a large, comprehensive support community for complicated steps.

7.6. AR Application Development

In this section, we would be discussing the steps in creating the AR application for this thesis. The tools used for the development are mentioned below:

1. Unity Editor version 2018.4 with Vuforia SDK
2. Visual studio 2017.
3. Inscape for material design.
4. Android Studio.
5. Account in vuforia development portal.
6. Test smart-device

The tools are selected as they all offer free options for student, easy to use and consume less resources while developing our AR application. We previously reviewed various smart devices or hosting device and after careful consideration, we decided the Xiaomi MI Mix 2S [#Xiaomimobile] would be the best hosting device for our application. solving the problem of selecting the best and capable hosting device

We as well previously reviewed various development engine to be used and based on the authors experience we decided to use Unity 3D development engine as it is a good option for developing mobile AR application, it also has a huge support community which helps is solve the problem of selecting the right development engine for our application.

To successfully develop a working prototype we followed the steps below:

7.7. Target Image Capturing

As we would be using the Marker-based method, we would need a good pictorial representation for our AR camera. Taking an image from the hosting device itself and uploading it to Vuforia for image recognition would be the right process to follow, but we had a problem of our AR camera not identifying our target object in the real world due to either change in lighting condition, unstable mobile placement or changes around the target object in the real world. To solve this problem we propose a method of first converting our hosting device simple camera to an AR camera. to do this we developed and deploy a simple AR camera application to the mobile phone without any virtual object rendering, we then proceeded to view the target image through the AR camera and then take a picture of the target object as seen through the AR camera itself. This method was tested with other various smaller application and it helps us solved the unstable image recognition problem, our final application was able to identify the target objects more efficiently and the virtual image projected are better quality for the end-user to interact with compared to the simple mobile camera.



Figure 5. Simple Mobile Camera.

After getting a good target image of our Multi-tank system, we then upload this image in our Vuforia developer database as we would be using Vuforia image recognition



Figure 6. AR Camera

algorithm to enable image recognition on our AR camera. We upload the image and Vuforia engine gives a rate for the best augmentable image and based on the rating we select the best augmentable target image for our AR application, the image is then downloaded as a unity package to be imported into Unity 3D assets.

The screenshot displays the Vuforia Engine developer portal. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. Below this, a green bar contains 'License Manager' and 'Target Manager'. The main content area shows a breadcrumb trail: 'Target Manager > Master-thesis > IMG_20200426_161024'. The target image is titled 'IMG_20200426_161024' and has a rating of 5 stars. The metadata includes: Type: Single Image, Status: Active, Target ID: 039430b72a6b46fa26d19d114d5f2, Augmentable: 5 stars, Added: Apr 26, 2020 16:12, and Modified: Apr 26, 2020 16:12. The image itself shows a yellow metal frame with yellow markers overlaid. At the bottom, there is a footer with links for PTC, Inc., Resources, Community, and AR Products, along with a newsletter sign-up form.

Figure 7. Vuforia Engine Image Rating.

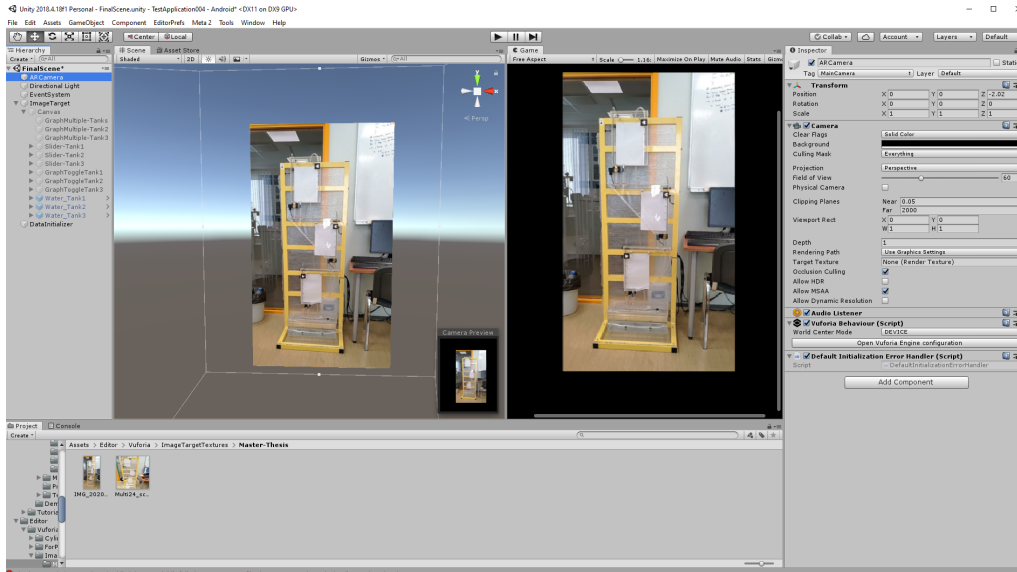


Figure 8. Target Image In Unity 3D

7.8. Virtual Object Integration

Mixed Reality is a powerful concept that can in a way help us experience more beyond our imagination and give us much more experience our human eyes can give. It is very essential for the developers of mobile AR application to understand that poor execution and design can in a way impact the end-users engagement, reaction and ultimately the way the end-user utilizes the application. AR application requires the integration of virtual object with the real environment and when developing AR application for smart devices such as our smartphone in which the screen size is already a constraint it would be beneficial for the developer to consider some of the points below

1. Size of the virtual object.
2. The position of the virtual object.
3. The material and quality of the virtual object.
4. The resemblance of the virtual object to the real object in the end-users real environment .

Some of the best Mobile AR application development best practice are[52]

1. Making sure the gameobjects are not obstructing the end-users view.
2. Making sure the user feels he/ she is in control of what they see.
3. Making virtual object interactions as simple and possibly automating them so the user does not have to perform much manipulations.

By taking some of these considerations into focus would help develop a good user-friendly AR application. Some of the development Engine does support or has the option of creating the needed virtual objects within the same development Engine. There are other third-party tools or software that can be used to create 3D objects for AR applications, some of the tools are namely:

1. SketchUP
2. Blender
3. SelfCAD
4. Inkscape

There are many more advanced tools for professional 3D virtual objects. For this thesis, we decided to make the best of the AR application we would use the Unity 3D game object design option while using Inkscape as a designing option for the materials which are to be imported into unity and integrated with the designed 3D game objects in Unity. Using third party applications mentioned above does help some of the development challenges but also cause some other problems affecting the optimization of our AR application mentioned below:

1. Application file size.
2. Application compilation.
3. File conversion.
4. Texture and virtual object quality.

To solve the problem, we decided designing the materials of the virtual objects on Inkscape and importing them either as a simple image or unity package would be the best option as it saves time, storage space and increase compilation time and reduce development time.

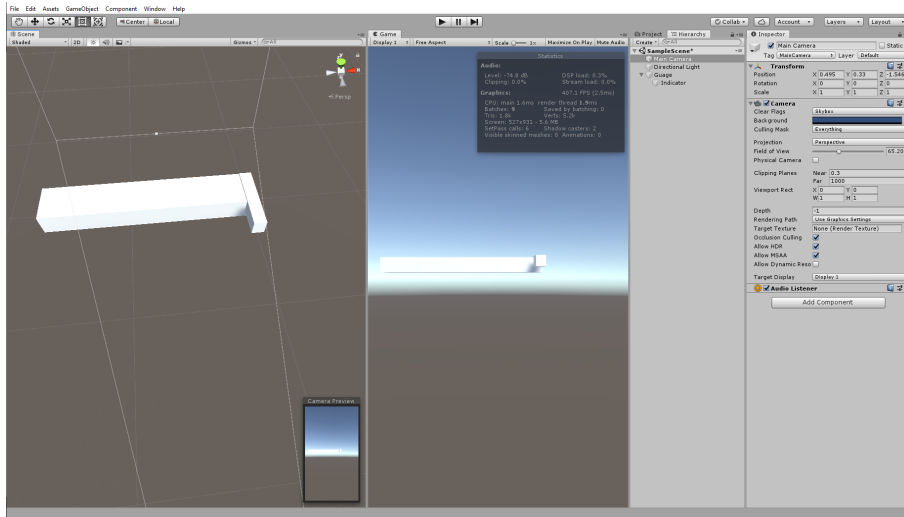


Figure 9. Unity 3D VO Without Material

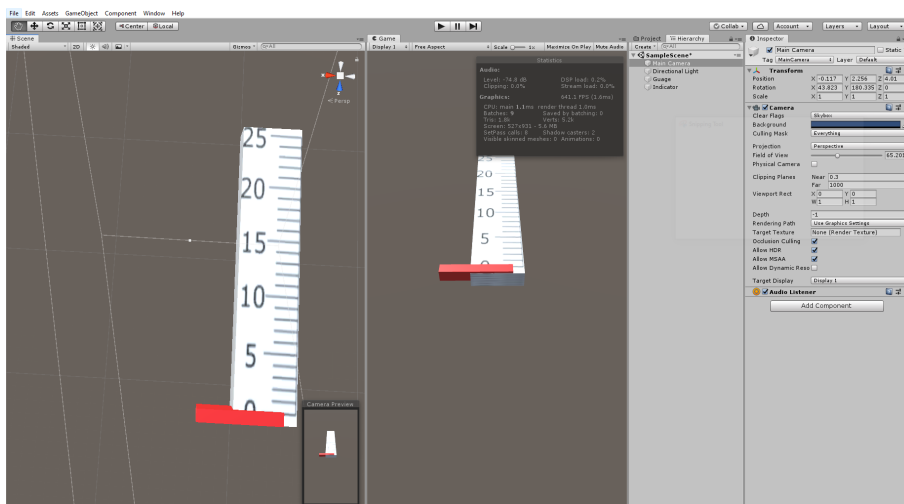


Figure 10. Unity 3D V.O With Material

After the designing of the virtual object, we then add the virtual object to our scene and position it close to the tank on the target image, so when our AR camera identifies our target object in the real world, it would overlay the virtual object for the end-user to interact with when deployed to the hosting device.



Figure 11. Overlaid Virtual Object

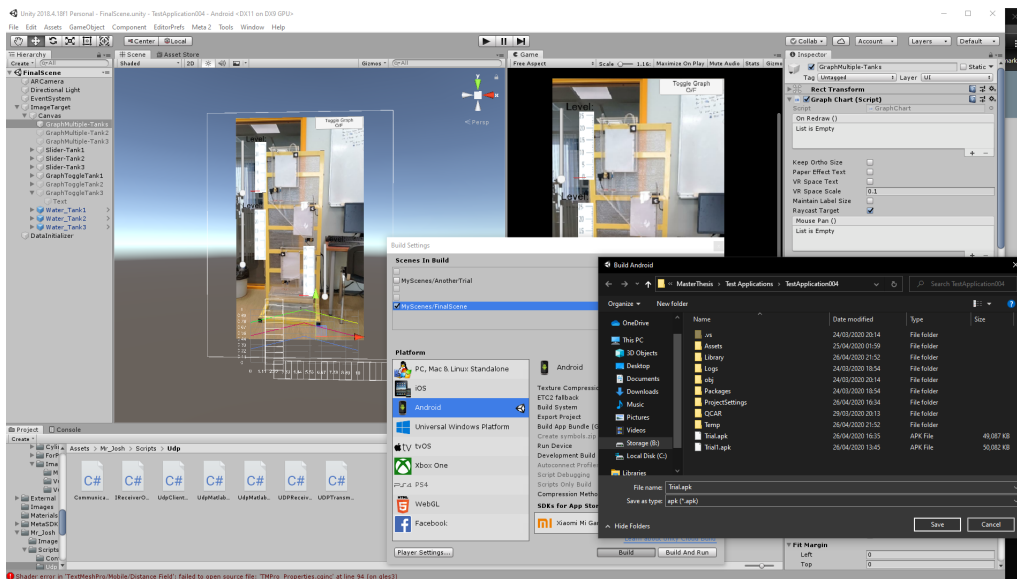


Figure 12. APK Deployment

7.9. Coupling Of All Gameobjects and Assets

Upon integrating our virtual object with the target image, we had a problem of ensuring the virtual object are position in the right place. AR application requires virtual objects and in this thesis, there are numbers of virtual objects in the application. Coupling of the virtual objects can be done in two ways namely

1. Parent-Child relationship
2. Scripting of the objects.

In our thesis, we utilized both options as objects in the application interact differently and behave differently. The parent-Child relationship works by attaching game objects in a hierarchy format so whatever happens to the parent object affect the child object. Scripting of the objects is done by attaching a C# script to the game object and specifying the type of behaviour we want the game object to have in form of a C# function by accessing the Unity game object properties.

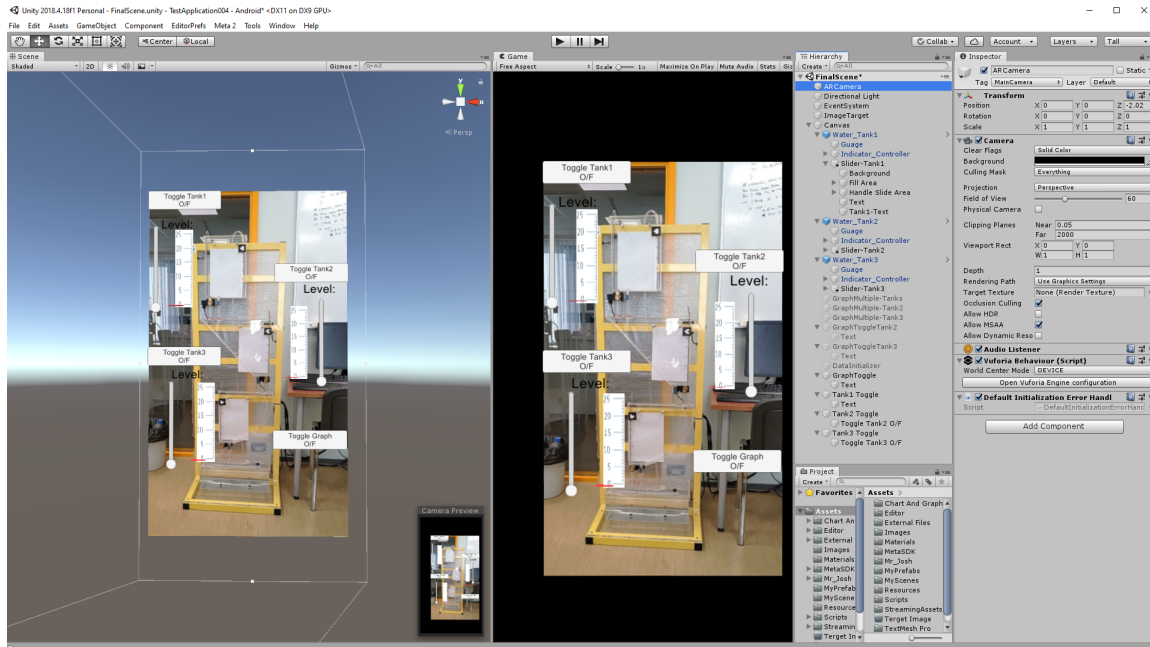


Figure 13. Coupling Of Game-object

After integrating our virtual object we then once again deploy it to our hosting device and then make sure all of the virtual objects, objects in the canvas are positioned well and responds as they should

7.10. Connecting The Multitank System To Unity

As our AR application is a standalone application, a remote access connection would be the only possible means of connecting and sending instructions to the Multi-tank system through Simulink. To establish this communication we would be using a C# script for the back end scripting of specifying the receiving IP address and the transmitting IP address along with the individual port, within our XR Lab as the applications and Simulink are of the same network range. We would be using Unity Transport Layer API working on top of the operating system's sockets-based networking, we then send and receive messages represented as arrays of bytes using a UDP networking protocol for full-duplex communication between our stand-alone application and Simulink. The user interacts with the virtual object on the hosting device, the instructions are passed in the form of bytes and sent to Simulink as inputs/instructions then processed to move the liquid level in the multi-tank in realtime. We designed a model on Matlab which is to interpret instructions from the AR application running on our hosting machine. In our XR lab, we had an existing model which is used to control the multi-tank system we simply specify the IP address and remote Port address for the Inputs packet as well as the IP address and Remote port for the Output packet.

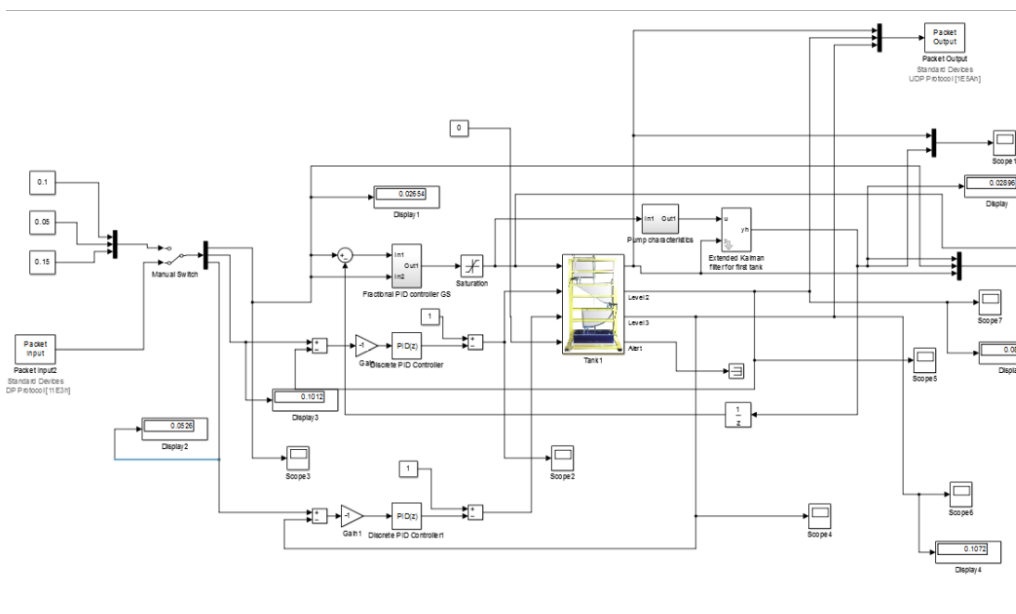


Figure 14. PID-Controller Model

After updating specified ports and IP addresses we then deploy the AR application once again and test to ensure the manipulated virtual object does control the

water-level in the real multitank system

7.11. Visualizing Data Between Communicating System

Data plays an important role in the developments and decision-making process, to visualize the data-stream between the three tanks, we utilized the scope view from our Simulink model, doing this we were able to find out the reachable points in the tank and which are not reachable as the second tank and third tank all depend on the first tank as this is where the liquid intake comes in from, to solve this problem of floating points/value between the tanks we propose a solution that each tank has its own pump so we can have definite control of each tank thereby eliminating the dependencies between the tanks. Upon deployments, we made series of test to find out and put together which set marks are reachable and which are not and upon the completion of the testing, we propose the multi-tank system needs recalibration and as mentioned before the ultimate solution for the dependencies between the tank is to have an individual pump for each separate tank.

7.12. Functionality And Use-Case Overview

Data plays an important role in the developments and decision-making process, to visualize the data-stream between the three tanks, we utilized the scope view from our Simulink model, doing this we were able to find out the reachable points in the tank and which are not reachable as the second tank and third tank all depend on the first tank as this is where the liquid intake comes in from, to solve this problem of floating points/value between the tanks we propose a solution that each tank has its own pump so we can have definite control of each tank thereby eliminating the dependencies between the tanks. Upon deployments, we made series of test to find out which AR application consists of the various interactable virtual object with different functions, We designed and positioned the virtual gauge which consists of set marks and an indicator for end-user to visualize the liquid level on the screen when not looking at the multi-tank system. We also utilized a Slider in the Unity 3D canvas, the slider is attached to the gauge and configured to control the liquid level indicator attached the gauge thus giving the end-user visual feedback on what level the liquid in the tank is. The value of the slider ranging from 0 to 25 is then passed

to Simulink for the real-world control of the water level in the multi-tank system.

Augmented Reality has many advantages and it can be applied in many areas depending on the functionality it is designed to perform. The technology has been explored in areas such as education, film making, military training and equipment servicing, Gaming, Storytelling and many more. we designed our AR application to control a system in our XR lab which gives us the functionality of remote access control, as this is the first mobile AR application dedicated designed for a control system in our lab, we propose more AR application with more functionality to control multiple control systems in our XR lab. In cases where there is a shortage of AR equipment such as HMD, controllers etc., mobile AR application can solve this problem, thereby giving the possibilities for anyone working in our XR lab a chance to explore from the comfort of their smart device of choice.

AR application such as ours can be used in industries where remote access controls are needed as workers might not be able to reach certain places, lift certain machinery or to automate processes we propose such AR application would be ideal for automated process controls.h set marks are reachable and which are not and upon the completion of the testing, we propose the multi-tank system needs recalibration and as mentioned before the ultimate solution for the dependencies between the tank is to have an individual pump for each separate tank.

7.13. Packets Exchange Security

Ensuring security is a paramount aspect of technology that should be taken seriously. There are many ways to ensure security for different kinds of systems and applications. For AR applications there are not much research conducted to ensure that the data flows between communicating devices are protected from unwanted parties. As this technology is still developing it would be good to ensure that there are no cyberattacks and ensuring loopholes within the applications/systems are well layered. As for our thesis, we conducted this test in our laboratory and as there are limited connected device to the network used in the lab we can say there are no un-solicited intrusions within our network however we considered that the use of Datagram Transport Layer Security [53] can help us ensure that the packets we send to Simulink and receive from Simulink are to some degree secure from external parties which could perform malicious actions such as Man in the middle attacks, packets flooding and many

more, these possible scenarios would cause catastrophic damages if it were to happen in real use case scenarios. As Matlab primarily does not support this feature, we speculate a loopback translation Engine could help solve this problem.

8. Conclusion

We at the onset of this thesis had the aim of creating a mobile AR application connected to a control system for remote access control. We considered various methods and options to best achieve this goals. As we progressed in research and design we were able to overcome challenges and choose the best available resources to help complete this thesis. As Mobile Augmented reality approach has not been adopted for the control of control systems in our lab before, we initially had questions such as:

1. Which Framework to use.
2. Which method to adopt.
3. Which hosting device to utilize and the what specifications we need.
4. How do we design the application.
5. How do we connect the systems.

We reviewed various options and based on the resources we were able to

1. Select a Framework for development.
2. Design the optimal virtual object for the user to interact with.
3. Select a method for best AR experience.
4. Select capable hosting device.
5. Deploy the application and use in the lab.

We further propose more development of mobile AR application in our AR lab as it is much cheaper, portable and almost as powerful as our normal HMD gears. However, there are still other topics which could be possible another direction or a challenge to be solved in future research or can be improved, area such as the security measures for mobile AR application and as mentioned in sub-section 7.13, as it is essential to ensure AR applications are secured from un-solicited intrusions.

Upon completion of this thesis, we are able to identify further topics for further research and propose more use case in using mobile AR application in controlling other control systems in our XR lab. the thesis has contributed to the personal developments and experience of the author on how to to utilize resources and understanding the technology behind AR.

References

- [1] J. Jerald, *The VR Book*. Association for Computing Machinery, oct 2015.
- [2] D. L. Baggio, S. Emami, D. M. Escriv , K. Ievgen, N. Mahmood, J. Saragih, and R. Shilkrot, *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing Ltd., 2012. [Online]. Available: <https://www.packtpub.com/application-development/mastering-opencv-practical-computer-vision-projects#tab-label-product.info.authors.tab>
- [3] A. M. Nazri and D. R. . A. Rambli, “Current limitations and opportunities in mobile augmented reality applications,” in *2014 International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, jun 2014. [Online]. Available: <https://ieeexplore.ieee.org/document/6868425>
- [4] C. America, “The history of augmented reality,” *Augmented Reality*, 2018. [Online]. Available: <https://www.colocationamerica.com/blog/history-of-augmented-reality>
- [5] P. F. Georgel, “Is there a reality in industrial augmented reality?” in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, oct 2011.
- [6] D. Wagner and D. Schmalstieg, “History and future of tracking for mobile phone augmented reality,” in *2009 International Symposium on Ubiquitous Virtual Reality*. IEEE, jul 2009.
- [7] A. A. Hopkins, *Magic: Stage Illusions and Scientific Diversions, Including Trick Photography.*, A. A. Hopkins, Ed. J. J. Little & Co., New York City, 1897. [Online]. Available: <https://www.gutenberg.org/files/45235/45235-h/45235-h.htm>
- [8] O. Merhi, E. Faugloire, M. Flanagan, and T. A. Stoffregen, “Motion sickness, console video games, and head-mounted displays,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 49, no. 5, pp. 920–934, oct 2007.
- [9] A. Kinsella, R. Mattfeld, E. Muth, and A. Hoover, “Frequency, not amplitude, of latency affects subjective sickness in a head-mounted display,” *Aerospace Medicine and Human Performance*, vol. 87, no. 7, pp. 604–609, jul 2016.

- [10] A. Ufkes and M. Fiala, “A markerless augmented reality system for mobile devices,” in *2013 International Conference on Computer and Robot Vision*. IEEE, may 2013.
- [11] M. Tikander, “Development and evaluation of augmented reality audio systems,” pp. Verkkokirja (1276 KB, 70 s.), 2009. [Online]. Available: <http://urn.fi/URN:ISBN:978-952-248-153-5>
- [12] K. Lyons, M. Gandy, and T. Starner, “Guided by voices: An audio augmented reality system,” 07 2000.
- [13] T. Palladino, “Google prepares to ship web-based ar support in chrome 81,” Tech. Rep., 2020. [Online]. Available: <https://next.reality.news/news/google-prepares-ship-web-based-ar-support-chrome-81-0247038/>
- [14] D. Ashaver and S. M. Igyuve, “The use of audio-visual materials in the teaching and learning processes in colleges of education in benue state-nigeria,” *IOSR Journal of Research & Method in Education (IOSRJRME)*, vol. 1, no. 6, pp. 44–55, 2013.
- [15] G. Inc., “Arcore developers overview.” [Online]. Available: <https://developers.google.com/ar/reference>
- [16] M. Kalia, P. Mathur, N. Navab, and S. E. Salcudean, “Marker-less real-time intra-operative camera and hand-eye calibration procedure for surgical augmented reality,” *Healthcare Technology Letters*, vol. 6, no. 6, pp. 255–260, dec 2019.
- [17] C. Sandor, T. Kuroki, S. Uchiyama, and H. Yamamoto, “Exploring visuo-haptic mixed reality,” 01 2007.
- [18] H. Ling, “Augmented reality in reality,” *IEEE MultiMedia*, vol. 24, no. 3, pp. 10–15, 2017.
- [19] S. Yonemoto, “Seamless annotation display for augmented reality,” in *2013 International Conference on Cyberworlds*. IEEE, oct 2013.
- [20] J. C. Chung, M. R. Harris, F. P. Brooks, H. Fuchs, M. T. Kelley, J. Hughes, M. Ouh-young, C. Cheung, R. L. Holloway, and M. Pique, “Exploring virtual worlds with head-mounted displays,” in *Three-Dimensional Visualization and Display Technologies*, S. S. Fisher and W. E. Robbins, Eds. SPIE, sep 1989.

- [21] M. R. Marnier, B. H. Thomas, and C. Sandor, “Physical-virtual tools for spatial augmented reality user interfaces,” in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, oct 2009.
- [22] M. Tarasevich, “Virtual objects integration and motion recognition for augmented reality applications,” Master’s thesis, Tallinn university of technology, 2016.
- [23] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent advances in augmented reality,” *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001.
- [24] N. Navab, “Developing killer apps for industrial augmented reality,” *IEEE Computer Graphics and Applications*, vol. 24, no. 3, pp. 16–20, may 2004.
- [25] A. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis, “Augmented reality applications in design and manufacturing,” *CIRP Annals*, vol. 61, no. 2, pp. 657–679, 2012.
- [26] J. Tumler, F. Doil, R. Mecke, G. Paul, M. Schenk, E. A. Pfister, A. Huckauf, I. Bockelmann, and A. Roggentin, “Mobile augmented reality in industrial applications: Approaches for solution of user-related issues,” in *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, sep 2008.
- [27] S. Goose, S. Sudarsky, X. Zhang, and N. Navab, “Speech-enabled augmented reality supporting mobile industrial maintenance,” *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 65–70, jan 2003.
- [28] J. Platonov, H. Heibel, P. Meier, and B. Grollmann, “A mobile markerless AR system for maintenance and repair,” in *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, oct 2006.
- [29] A. Rohacz and S. Strassburger, “Augmented reality in intralogistics planning of the automotive industry : State of the art and practical recommendations for applications,” in *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*. IEEE, apr 2019.
- [30] O. Blanco-Novoa, T. M. Fernandez-Carames, P. Fraga-Lamas, and M. A. Vilar-Montesinos, “A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard,” *IEEE Access*, vol. 6, pp. 8201–8218, 2018.

- [31] X. Qiao, P. Ren, S. Dustdar, L. Liu, H. Ma, and J. Chen, “Web AR: A promising future for mobile augmented reality—state of the art, challenges, and insights,” *Proceedings of the IEEE*, vol. 107, no. 4, pp. 651–666, apr 2019.
- [32] X. Qiao, P. Ren, S. Dustdar, and J. Chen, “A new era for web AR with mobile edge computing,” *IEEE Internet Computing*, vol. 22, no. 4, pp. 46–55, jul 2018.
- [33] P. Dangkham, “Mobile augmented reality on web-based for the tourism using HTML5,” in *2018 International Conference on Information Networking (ICOIN)*. IEEE, jan 2018.
- [34] *Open AR Cloud*. [Online]. Available: <https://www.openarcloud.org>
- [35] PTC, *Vuforia Studio*. PTC Inc., 2019. [Online]. Available: <https://www.ptc.com/en/products/augmented-reality/vuforia-studio>
- [36] A. Inc., “Iphones,” 2020. [Online]. Available: <https://www.apple.com/iphone/>
- [37] —, “Apple iphone x,” 2020. [Online]. Available: <https://www.apple.com/shop/buy-iphone/iphone-x>
- [38] S. E. Inc., “Galaxy s9 and s9+,” 2020. [Online]. Available: <https://www.samsung.com/global/galaxy/galaxy-s9/>
- [39] S. Inc., “Specification samsung galaxy s9 & s9+.” [Online]. Available: <https://www.samsung.com/global/galaxy/galaxy-s9/specs/>
- [40] A. C. Inc., “Zenfone ar (zs571kl),” 2017. [Online]. Available: <https://www.asus.com/Phone/ZenFone-AR-ZS571KL/>
- [41] A. Inc., “Asus zenfone ar (zs571kl).” [Online]. Available: <https://www.asus.com/us/Phone/ZenFone-AR-ZS571KL/Tech-Specs/>
- [42] G. Inc., “Google pixel slate.” [Online]. Available: https://store.google.com/us/product/pixel_slate
- [43] —, “Google pixel slate.” [Online]. Available: https://store.google.com/us/product/pixel_slate_specs
- [44] A. Inc., “Apple ipads.” [Online]. Available: <https://www.apple.com/>
- [45] —, “Apple ipads.” [Online]. Available: <https://www.apple.com/ios/augmented-reality/>

- [46] X. Inc., “Mi mix 2s overview.” [Online]. Available: <https://www.mi.com/global/mix2s>
- [47] *Xiaomi MI Mix 2S*. [Online]. Available: <https://www.mi.com/global/mix2s/specs>
- [48] *AR Core overview*. [Online]. Available: <https://developers.google.com/ar/discover>
- [49] *Unity 3D*. [Online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview>
- [50] P. ACE, “Wheel replacement training,” *Mixed Reality*, 2017. [Online]. Available: <https://thinkmobiles.com/blog/augmented-reality-companies/>
- [51] P. Inc., “An overview of unreal engine,” *Mixed Reality*, 2015. [Online]. Available: <https://hub.packtpub.com/overview-unreal-engine/>
- [52] U. 3D, “Best practices for bringing ar applications to the field,” *Augmented Reality*, 2020. [Online]. Available: <https://blogs.unity3d.com/2020/02/28/best-practices-for-bringing-ar-applications-to-the-field/>
- [53] M. Nagendra and R. Eric, “Datagram transport layer security version 1.2,” *Internet Engineering Task Force (IETF)*, 2012. [Online]. Available: <https://tools.ietf.org/pdf/rfc6347.pdf>

A. Appendix 1. C# script

Transmitter:

```
using ChartAndGraph;

using System;

using System.Net;

using System.Net.Sockets;

using UnityEngine;

using UnityEngine.Experimental.UIElements;

public class UDPTransmitter : MonoBehaviour

{

    //Gameobjects

    public GameObject Water_Indicator1;

    public GameObject Water_Indicator2;

    public GameObject Water_Indicator3;

    public GameObject Meter_Gauge1;

    public GameObject Meter_Gauge2;

    public GameObject Meter_Gauge3;

    private Vector3 startPosition1;

    private Vector3 startPosition2;

    private Vector3 startPosition3;
```

```

[Range(0, 25)] public float Water_Level1;

[Range(0, 25)] public float Water_Level2;

[Range(0, 25)] public float Water_Level3;

//Slider functions

public GameObject Slider_touch_1;

public GameObject Slider_touch_2;

public GameObject Slider_touch_3;

//IP address

public string IP;

public int TransmitPort;

private IPEndPoint _RemoteEndPoint;

private UdpClient _TransmitClient;

private float Timer = 1f;

private float X = 4f;

public GraphChart chart;

double[] simulink_Levels_Array;

private void Start()

{

//Makes the indicator stick to the guage, thereby completing the guage

startPosition1 = Meter_Gauge1.transform.position;

startPosition2 = Meter_Gauge2.transform.position;

```



```

startPosition3 = Meter_Gauge3.transform.position;

Initialize();

}

private void Update()

{

//Water tank float newWaterLevel1 = Water_Level1 / 2.5f;

Water_Indicator1.transform.localPosition = new Vector3(startPosition1.x, newWaterLevel1, startPosition1.z);

float simulink_level1 = (Water_Level1 / 100);

float newWaterLevel2 = Water_Level2 / 2.5f;

Water_Indicator2.transform.localPosition = new Vector3(startPosition2.x, newWaterLevel2, startPosition2.z);

float simulink_level2 = (Water_Level2 / 100);

float newWaterLevel3 = Water_Level3 / 2.5f;

Water_Indicator3.transform.localPosition = new Vector3(startPosition3.x, newWaterLevel3, startPosition3.z);

float simulink_level3 = (Water_Level3 / 100);

simulink_Levels_Array = new double[] { simulink_level1, simulink_level2, simulink_level3};

Send(simulink_Levels_Array);

//Adding data stream. i.e where the data is coming from to feed the graph.

Timer -= Time.deltaTime; //so each data update, we decrease the time that has passed.

if (Timer <= 0f)

```

```

{

Timer = 1f; //set the timer to 1 and:

chart.DataSource.AddPointToCategoryRealtime("Tank1", X, UnityEngine.Random.value,
1f); //setting it to 1f to blend the animation to 1s

chart.DataSource.AddPointToCategoryRealtime("Tank2", X, UnityEngine.Random.value,
1f);

chart.DataSource.AddPointToCategoryRealtime("Tank3", X, UnityEngine.Random.value,
1f);

X++; //increase the X vlaue so the nextpoint is about 1 unit away from the previous
one.

}

}

//Slider function

public void Slider_Moved_tank1(float newValue)

{

foreach (Touch touch in Input.touches)

{

//Activate the users finger touch on the screen// only for the Sliders

if ((touch.phase == TouchPhase.Began) || (touch.phase == TouchPhase.Moved ||
(Input.GetMouseButtonDown(0))))

{

Vector3 pos = Slider_touch_1.transform.localPosition;

Water_Level1 = newValue; pos.y = newValue / 2.5f;

Slider_touch_1.transform.localPosition = pos;

```

```

}

}

}

//Initializing remote connection on application start

private void Initialize()

{

_RemoteEndPoint = new IPEndPoint(IPAddress.Parse(IP), TransmitPort);

_TransmitClient = new UdpClient();

}

/// <summary>

/// Sends a double value to target port and ip.

/// </summary>

/// <param name="val"></param>

public void Send(double val)

{

try

{

// Convert string message to byte array.

byte[] serverMessageAsByteArray = BitConverter.GetBytes(val);

//byte[] data = Encoding.UTF8.GetBytes(val);

_TransmitClient.Send(serverMessageAsByteArray, serverMessageAsByteArray.Length,

```

```

_RemoteEndPoint);

}

catch (Exception err)

{

Debug.Log("<color=red>" + err.Message + "</color>");

}

}

/// <summary>

/// Sends a double array to target port and ip.

/// </summary>

/// <param name="val"></param>

public void Send(double[] val)

{

try

{

for (int i = 0; i < val.Length; i++) Send(val[i]);

}

catch (Exception err)

{

//Debug.Log("<color=red>" + err.Message + "</color>");

}

}

```

```

}

//On application close

/// <summary>

/// Deinitialize everything on quitting the application.Or you might get error in
restart.

/// </summary>

private void OnApplicationQuit()

{ try

{

    _TransmitClient.Close();

}

catch (Exception err)

{

    Debug.Log("<color=red>" + err.Message + "</color>");

}

}

}

}

//Receiver

using UnityEngine;

using System;

using System.Net;

```

```

using System.Net.Sockets;

using System.Threading;

public class UDPReceiver : MonoBehaviour
{
    string RemoteAddress = "127.0.0.1";

    public int PortRemote = 3200;

    private UdpClient _ReceiveClient;

    private Thread _ReceiveThread;

    private IReceiverObserver _Observer;

    void Start()
    {
        Initialize();
    }

    /// <summary> /// Initialize objects. /// </summary>
    public void Initialize()
    {
        // Receive _ReceiveThread = new Thread(new ThreadStart(ReceiveData));
        _ReceiveThread.IsBackground = true; _ReceiveThread.Start();
    }

    public void SetObserver(IReceiverObserver observer)
    {
        _Observer = observer;
    }
}

```

```

}

/// <summary> /// Receive data with pooling. /// </summary>

private void ReceiveData()

{

    _ReceiveClient = new UdpClient();

    while (true)

    {

        try

        {

            IPEndPoint IP = new IPEndPoint(IPAddress.Any, PortRemote);

            byte[] data = _ReceiveClient.Receive(ref IP);

            double[] values = new double[data.Length / 8];

            Buffer.BlockCopy(data, 0, values, 0, values.Length * 8);

            Debug.Log(values);

            if (_Observer != null)

                Debug.Log(">>>>");

        } catch (Exception err)

        {

            Debug.Log("<color=red>" + err.Message + "</color>");

        }

    }

}

```

```

}

/// <summary> /// Deinitialize everything on quitting the application.Or you might
get error in restart. /// </summary>

private void OnApplicationQuit()

{

try

{

_ReceiveThread.Abort();

_ReceiveThread = null;

_ReceiveClient.Close();

} catch

(Exception err)

{

Debug.Log("<color=red>" + err.Message + "</color>");

}

}

}

```