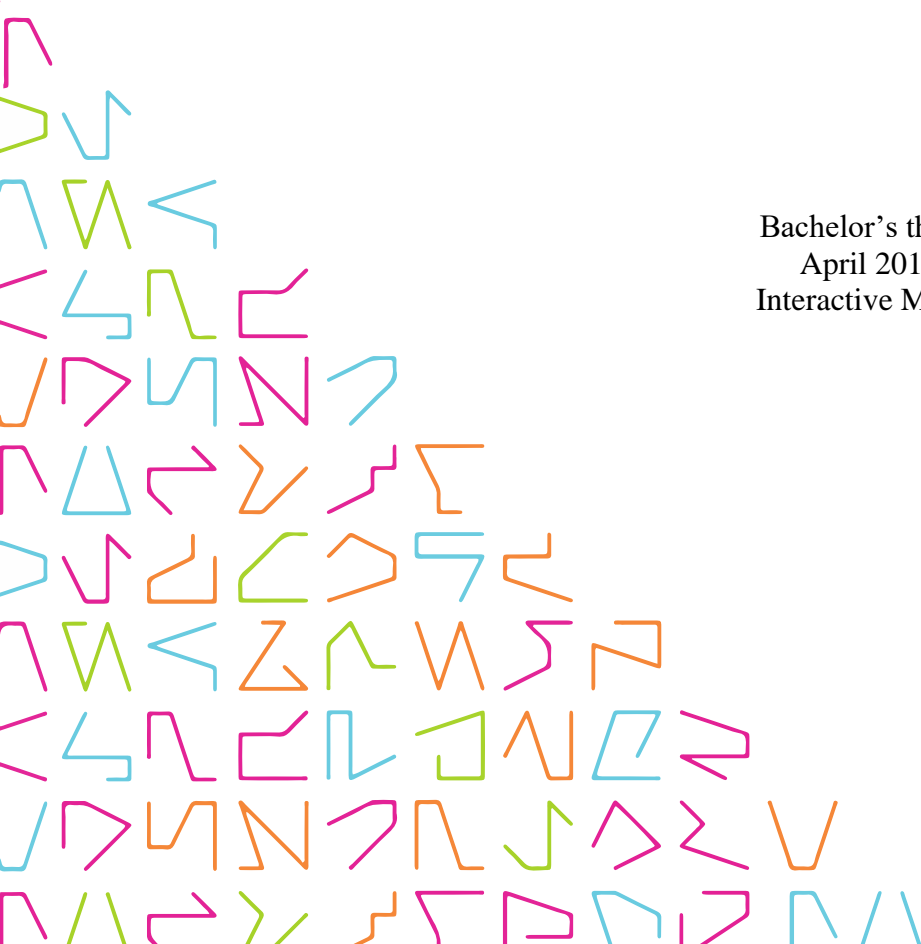


# **Creation of an Augmented Reality App for an Introduction to Industrial Ma- chine Mechanics**

Zachary Compton

Bachelor's thesis  
April 2018  
Interactive Media



## **ABSTRACT**

Tampere University of Applied Sciences  
Interactive Media

Zachary Compton:  
Creation of AR App for an Introduction to Machine Mechanics

Bachelor's thesis: Pages 26, Appendices 2 pages  
April 2018

---

This paper describes the development of an Augmented Reality mobile application to help introduce the mechanics of machinery in an industrial environment. The basic aspects of Augmented Reality concepts along with a step-by-step process of developing an application will be offered. The need for good understanding on how a machine operates and its motions are excellent domains for Augmented Reality applications. The idea is to bring visual information of industrial machines as effortlessly as possible into the view of a user. The key hardware of this application will be a mobile phone, which uses an image detection system named Vuforia, that superimposes the augmentations of a working welding arm in the view of the user. Aspects in the usage of this application related to equipment introduction and providing an experience by accessing the inaccessible will be discussed.

## CONTENTS

1	INTRODUCTION.....	6
2	History of AR .....	7
2.1	Background of AR .....	9
2.1.1	Progression of AR software .....	10
2.1.2	How AR works.....	11
2.2	App Development .....	13
2.2.1	Graphical 3D Models .....	15
2.2.2	Animation for AR .....	17
2.2.3	Scene Creation in Unity .....	17
2.2.4	GUI Creation.....	19
3	Goal and Overview.....	22
3.1	Problems Raised with App Creation.....	23
3.1.1	Improving App Development .....	23
4	Conclusion.....	25
	REFERENCES.....	26
	APPENDICES .....	27
	Appendix 1. Unity Scripts .....	27
	A. 27	
	B. 28	

**ABBREVIATIONS AND TERMS**

AR	Augmented Reality
VR	Virtual Reality
HMD	Head mounted display
OS	Operating System
GUI	Graphical user interface
3D	Three Dimensional
LCD	liquid-crystal display
CAD	Computer-aided design
UI	User Interface
iOS	Apple Operating Software
Android	Android Operating Software
SDK	Software Development Kit
C#	Programming Language
SIG	Silicon Graphics, Inc.

## Table of Figures

<u>Figure 1 Peppers Ghost</u> .....	7
<u>Figure 2 Sword of Damocles</u> .....	8
<u>Figure 3 Super Cockpit</u> .....	9
<u>Figure 4 Fetus AR Projection</u> .....	10
<u>Figure 5 Unity 3D</u> .....	10
<u>Figure 6 CAD Model</u> .....	14
<u>Figure 7 Unity Integration</u> .....	17
<u>Figure 8 GUI</u> .....	18
<u>Figure 9 Particle</u> .....	19

## 1 INTRODUCTION

As virtual reality (VR) and Augmented Reality (AR) continue to introduce themselves in to consumer software and hardware markets, many industrial and mechanical companies have been slow to familiarize the technology to the factory facilities. This slow adoption of technology can likely be attributed to a lack of understanding and awareness of VR and AR technology, grouping it in to only being used for entertainment purposes. These technologies should not be grouped together, wherein they are used quite differently.

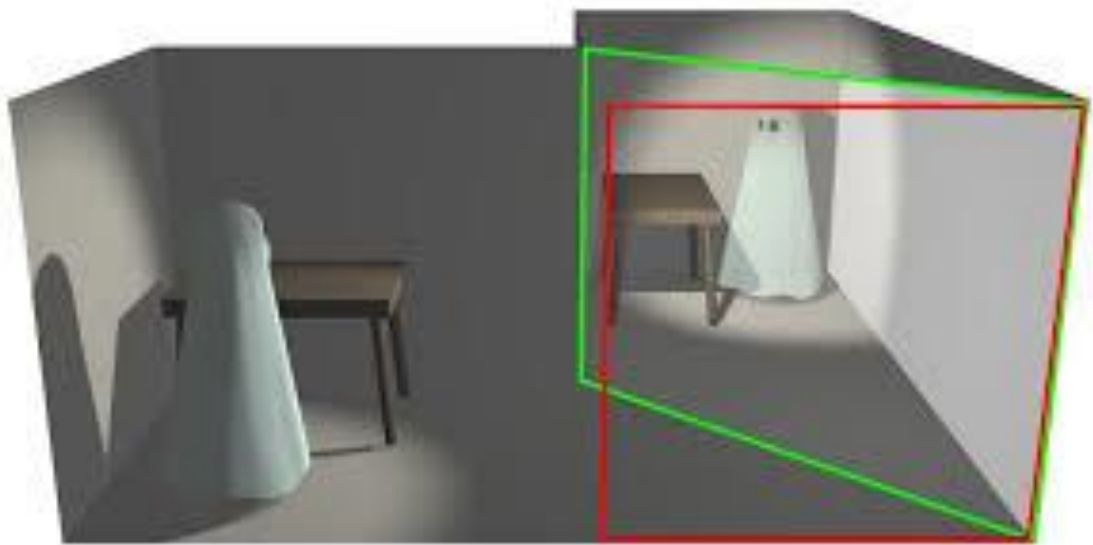
I wish to introduce the process of creating an AR app that could be used as an introduction into the industrial work environment and equipment. The app is developed in the gaming engine Unity with an AR plugin software; Vuforia. This software will allow the viewer to see the mechanical movement of factory equipment, specifically a robotic welding arm that normally would be quite dangerous to view while in use. The app operates on a mobile device such as a phone or tablet and can be used as an introduction to dangerous equipment for training or safety classes. It shows a graphical rendered automated welding arm contained in a generated room in place of the physical welding arm which would be non-operational or closed from view as a safety precaution to people in the general area. The user can interact with the graphical user interface (GUI) on the mobile device, being able to select an animation for the mechanical arm to demonstrate its welding capabilities and movements.

Over all the app is sight specific and mimics the physical welding arm and the welding process the arm goes through. Graphical AR information is provided directly into the view of the user through the mobile device, overlaying AR graphics over the physical environment. This action is triggered by an image located in the physical world, this image is scanned by the mobile device to activate and generate an AR environment. This is where the digital world will blend with the physical world.

AR is a great medium for situations where the viewer can get first person experience of equipment in use while not in a hazard position. The AR experience can help condition responses where correct behaviour is needed for industrial surroundings. This type of experience can give immediate influence and information for a project as well as expose the limitations of equipment and people.

## 2 History of AR

It's easy to go far back in time and find many examples of layering information on top of the physical world as such is the 'Pepper's Ghost' effect in 1862. (Brooker, 2007) Pepper's Ghost was created by Henry Dircks and it was an optical illusion that deployed visual effects using glass and light. It involved using a large pane of glass set at an angle between two rooms placed on a theatre stage, room one was oriented so that theatre viewers looked straight ahead into this room that was brightly lit. The second room, holding a "ghostly" scene, was hidden from view, dark and positioned so that it was reflected in the pane of glass. As the audience viewed forward the lights in the dark hidden room were slowly illuminated as the lights in the lit room were slowly dimmed. This caused the "ghostly" scene to be reflected in the pane of glass and appear to the audience.



(Figure 1) Peppers Ghost

This was one of the first examples of layering an image onto another environment.

The first working prototype of computer-generated AR was created by Ivan Sutherland in 1968. He created a head mounted system with an optical see-through display and a mechanical tracking system. His system had a 40° field of view, which enabled the user to see synthetic computer-generated information layered together over the viewers perception of the real world.



(Figure 2) Sword of Damocles



Also in development in 1968 the United States armed forces had its own research on AR, developing a program for the US Air Force. (Livingston) The 'Super Cockpit' as it was called was the prototype for the modern Heads-up display (HMD) that is still used by pilots today. The HMD was superimposing flight details to the pilot, as stated "The key feature of this system was providing spatial awareness for the pilots to understand and incorporate into his course of action a variety of incoming data streams." (Livingston) Not until 1992 do we get the actual term "AR" first used in the Boeing company for a process of simplifying assembly of wire bundles by showing the schematics in the production of Air planes through an HMD.



(Figure 3) Super Cockpit

AR is not a new phenomenon but has advanced in huge bounds within the last 50 years. Some of that fastest growth started in 2007-2008 with the presentation of the iPhone iOS and Android OS providing AR technology to the commercial market and the AR experience to move from military applications to the overall public.

## 2.1 Background of AR

AR became a unique field of study in between the 1980s and 1990s. Around the same time of Boeing's introduction of the HMD device, in 1993 the first handheld display with a liquid-crystal display (LCD) screen. As stated, 'The screen showed the video output of

a Silicon Graphics, Inc. (SGI) graphics workstation of the time and was spatially tracked using a magnetic tracking device. This system was capable of showing contextual information as the user moved the device around—for example, giving detailed information about a location on a wall-mounted map.’ (Hollerer, Schmalstieg)

At the University of North Carolina in 1994, AR was developed for a medical purpose of allowing a doctor to observe a fetus within a pregnant woman. Based on the times, it was not the most accurate or detailed image displayed but it gave new paths and invention of the power of AR.



(Figure 4) Fetus AR Projection

Free moving wearable AR computing was being established at the MIT Media Lab around the mid-1990s. Called the ‘WearCam project’ it consisted of a computer in a bag, strapped about the waist of the user with a video see-through HMD. The user had the capability to change and alter the view they saw through the HMD. This technology recognized the beginning of wearable computers aligning its self with AR. Then in 1995 with the fully operational Global Positioning Service (GPS), tracking was possible so the user could have computer-generated graphical images correctly oriented to the viewers location and gaze. (Global Positioning System)

### 2.1.1 Progression of AR software

The early 2000’s saw AR software named ARToolKit, released to the public, it was the first open-source software allowing the overall public to program and create apps for Augmented Reality. This software allowed a mobile device to track and produce a 3D object from a black and white image. Also, during this time in Germany the professional and industrial communities became aware of the possibilities of AR, thanks to the educational office in the German government, who enacted a 21-million-euro research program

to develop AR to use in an industrial environment. (Binzer) The advancement of cell phones in the 2000's allowed AR software and AR systems to develop into a full function computing device that could handle the computing power and tracking ability needed for AR apps. In 2008 one of the most natural and easy to use AR software and tracking system named Vuforia became commercially available for development that allows the consumer and industrial market to create and advance AR applications.

### **2.1.2 How AR works**

Starting with a quick explanation of VR can help visualize AR. VR creates a fully immersive computer-generated environment that a user can view and interact with using optical head gear and hand-held controls. Associating a created computer world and the real world, AR would be closer on a scale to the real world. AR adds computer graphics and sounds to the real world viewed through a device such as a cell phone, tablet or eye-wear. Let's imagine we have before us a small window frame that can be held in front of your field of view, you also have a transparent sheet of plastic that has a picture of a tree on it. If look through the window frame at an open field you will see grass, dirt, the sky and clouds, but there are no trees in your field of view. Now imagine taking the sheet of plastic and placing it over the window so that you must look through the transparent plastic and the window at the same time. In your field of view, you still see the grass, dirt, sky and clouds but by layering the image of the tree over the window you also see the tree as part of the scenery. To your view the tree looks as if it's in the field before you, a part of the whole scene. This layering effect is AR, layering generated images over the view of the natural world.

AR operations can be separated to two bases; location-based and recognition-based. Location-based uses GPS and a mobile devices internal gyroscope, this determines the location and orientation of the user so that the virtual information can be displayed in a appropriate position. The most common use of location-based AR uses Point of Interest (POI's), as an example a statue in Italy can be used as a POI and information can be generated as AR for a user based on the software's recognition of the statue. The second is recognition-based, where an object or an image can provide the wanted virtual information. In recognition-based AR there are three types of tracking used: marker tracking, markerless tracking and 3D tracking.

Marker tracking uses markers or bitmap images for the mobile camera to focus on. These images tend to be any kind of 2D high contrast image but most commonly used are QR-codes which are made up of black and white squares or designs.



(Figure 5) QR code

The black and white design helps determine the correct direction and perspective in which graphical content can be generated.

Markerless tracking includes printed images, such as pictures, advertisement images and poster, all of which are found in common public areas. These are different than the marker tracking as they don't have to be just black and white as in a QR-code. But they do have to contain a detail of uniqueness so that it can be recognized by the software without being mistaking for some other image. This can be in the form of high contrast points within the image allowing the ease of tracking.

3D tracking is the focussing and recognition of real world objects in 3D. This type of tracking is able to relate the real-world 3D object into the space of the generated graphical content and have the graphical content to be able to interact with the real-world object as if it is in the same vicinity.

To receive or generate a picture to layer over the real world an AR app would use a marker based image recognition. We create a marker or an image 'often a black and white bar-code image.' (Chavan, 2016) and save data into the application, while using the application on a mobile device that has a camera. When the camera sees the marker based image, then it brings up the information stored in the app. That data would be in the form of text or a 3D model and that data would be layered over the devices camera view of the real world.

## 2.2 App Development

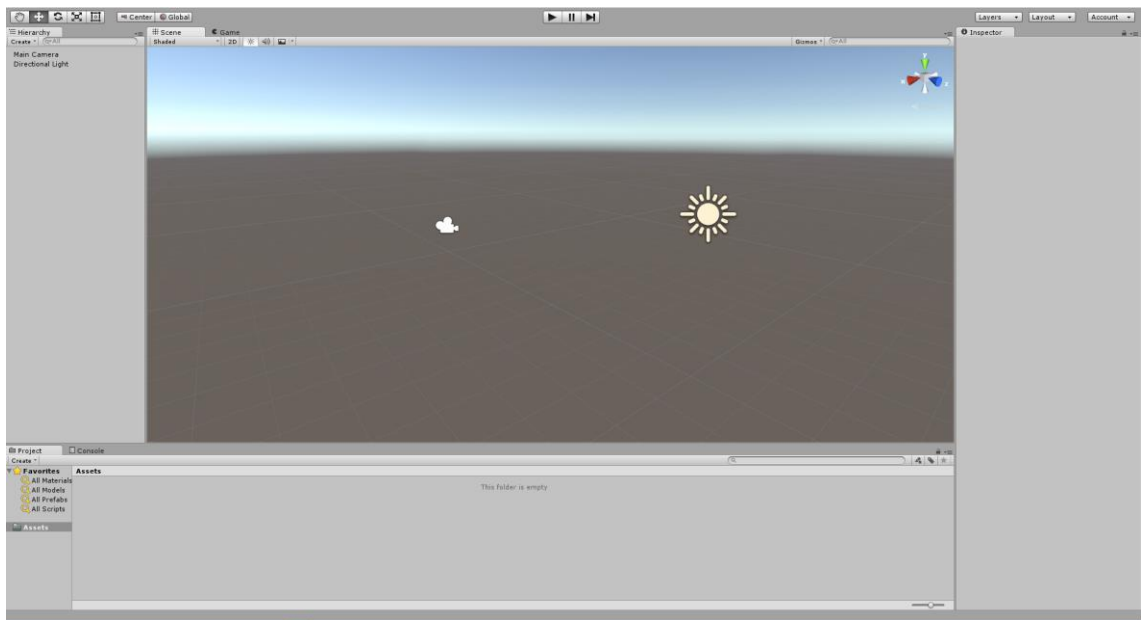
Depending on what kind of mobile application or ‘App’ a developer wants to create, they have multiple choices of AR tools to choose from. There are six main AR tool kits that can be used for Android or iOS platforms. The first is the ARLab that offers two AR software development kits (SDK) for iOS and Android, Image Matching and Point of Interest. Image matching, which recognizes and matches any image targets without the need to be connected to the internet and then there is Point of Interest (POI) recognition which is GPS based AR. (JulyRipid) Mataio is an AR SDK that was bought by Apple in 2015, this SDK supports 3D and 2D target tracking, face recognition and gesture detection along with GPS location tracking. Mataio is compatible with Android, iOS, PC and Unity using a target database that can work connected online or not. (JulyRipid) Another SDK that supports 2D and 3D is D’fusion, it also has gesture and face detection that is appropriate for development on Android and iOS. Wikitude is one AR SDK that is at the top of AR development, working with instant tracking and extended tracking, giving the ability to track images and objects beyond the view of the hardware’s camera. (JulyRipid) The final SDK is Vuforia, this kit enables the developer to create interactive applications that utilizes target tracking (marker, image, object, text), virtual buttons, extended tracking and smart terrain which detects real world surfaces. AR development requires intense software procedures and device management, much of which is handled by Vuforia. It was first developed by Daniel Wagner who was making a marker tracking system for mobile phones, as of this publication Vuforia supports devices from mobile phones to wearable eyewear and is one of the leading image tracking software available for AR app creation.

For AR app development, using a gaming platform will provide powerful graphics and a script editor that is driven by C# language programming for technical scripting. One of the best and most robust gaming platforms is Unity 3D, which will provide sufficient and up to date computer graphics for photorealistic rendering, animation for objects, physics and collision system with a easy user interface and input system. With Unity, a project can be created for any mobile platform, including Windows, iOS, Android and many others. For my app to generate and hold heavy amounts of data needed for the creation of AR, using the plugin software Vuforia will be most valuable to a new developer because its compatibility, ease of use and offers seamless integration with Unity.

To get started with Vuforia an account is needed and can be created in their Developer Portal website. The only information needed is an email and passcode that is generated

for the user by Vuforia. When the account is created and logged into Vuforia's website, a license key that is unique to one project is needed and can be created in the 'Developers' page on the website. Once all the information is filled, confirmation is needed, then on the 'License Manager' page a unique key number is generated to use later in Unity, it will need to be copied for later use. Next, a database and an image is needed for the image based target. An image that I previously made will be used as my marker target, previously presented in chapter 2.1.2.

This process of superimposing virtual objects happens by recognition of the image/marker target by the mobile phones camera and then rendering out the AR content. Once this image is recognized the AR software generates the virtual content. The platform to create this AR app with Vuforia is the game engine Unity.



(Figure 6) Unity 3D

The AR software Vuforia, supports a large range of targets from image targets to object targets, but image targeting will be more amicable for the environment the app is created for. Using a marker tracking pipeline, a bitmap image designed in photoshop is created for my app, specifically a black and white design will be exclusively used in this app, which is printed and posted in the industrial environment containing the mechanical welding arm. Once the image is ready to use it is upload to Vuforia and the size of my target image is entered in. Vuforia needs the size of the target image so that it knows how large the graphical content rendered out should be. Because of my limited printing capabilities, I will be using an A4 sized paper so my image is around 20 cm. After the image is

uploaded to the Vuforia website, it will be rated. This means the image must be unique enough and have enough contrasting points for tracking. Vuforia does this by giving a star rating, five stars is excellent and one star is barely functional. With a good rating, we can start building a scene with 3D models that will be created as AR in the app.

### **2.2.1 Graphical 3D Models**

Depending on the use of an AR app, all information presented to the user from the application can be pictures, videos, sounds, CAD models and 3D animations. With mechanical machinery it can be useful to see a 3D model of a complicated piece of equipment in conjunction with its virtual counterpart. This enables the user to see how the machine works and the ways it can be integrated to the user's activities. The attention of the user should be directed to the main elements of the apps production purpose, therefore an interactive AR workspace is created around the user enabling the user to feel at ease and comfortable with the information displayed. Some ideas for a strong connection with constructed 3D models and the user is to have semi-transparent elements for the overall appearance allowing distinction between virtual objects and real-world objects. When rendering objects in the app it's important to know that the viewers eyes are slightly offset based on the interpupillary distance (distance between the eyes). This offset is known as the parallax effect. This effect, if not properly dealt with can cause a disconnect between the rendered-out object and the real-world view. Unity includes scripts specific for this effect in its platform and allows VR and AR to be developed with in its game engine. Another difficult problem can be photorealistic rendering of 3D objects and how they are perceived in the real-world setting. This issue can be improved with correctly lighting the model and real-world textures. As part of Unity and Vuforia lighting is dealt with by the camera capturing the ambient light and adjusting the virtual scene lighting according to the real-world lighting. This helps with a realistic view of the 3D objects in the real world. (Linowes, Babilinski)

The models that will be computer-generated for my app are very specific piece of equipment that must be identical to the real-world welding arm. For this I obtained the computer-aided design (CAD) 3D model from Tampere University of Applied Science engineering department where the machine is held. CAD models are used for detailed model and design of products that will be built for manufacturing purposes. CAD models are too heavy of information to use in AR apps because of the polygon count of the models, so the CAD model should be cleaned up and information removed, essentially, rebuilding the model using the base model as a reference.



(Figure 7) CAD Model

To rebuild the 3D model, I use software specifically used to build and clean up 3D model files, 3Dsmax. I import my CAD model in to 3Dsmax and start to rebuild the model by deleting information like bolts and gears because these small models that won't been seen by the user and aren't useful to the app. After deletion of unneeded 3D models, I start to reduce the number of polygons in the models that will be used by looking at each model and deleting edges and vertices of polygons not needed. This is a tricky job because if too much information is deleted from the model it will lose its shape and detail and the viewer won't be able to identify the welding arm or the room and equipment around it. After a week of deleting and reducing the poly count the number of polygons in the model was reduced from 32 million to around 120,000 which is a significant reduction for my app.

Now that the 3D models, which consist of: A mechanical welding arm, the room it operates in, attachments for the welding arm and objects the welding arm will work on, are cleaned up and ready, textures are created to put onto the models so that they will be visibly similar to the real-world objects. UV maps are needed for each model so textures are placed the correct way on the 3D models.

UV mapping is a process of projecting a 2D image onto the surface of a 3D model. UV maps are created for each model in 3Dsmax by adding a modifier to each model that will take the 3D model, make a copy of it and unfold it into a 2D image. The 2D image is attached to the 3D model for textures, this is a way to control which pixels of my texture corresponds to the matching vertex on my 3D model. To create textures, I export my



models to Substance Painter, a texture creation software. Substance Painter gives the ability to paint colour and add detail to the models consequently allowing the models to look realistic. The textures, once created are exported out and ready to use with the 3D models in Unity.

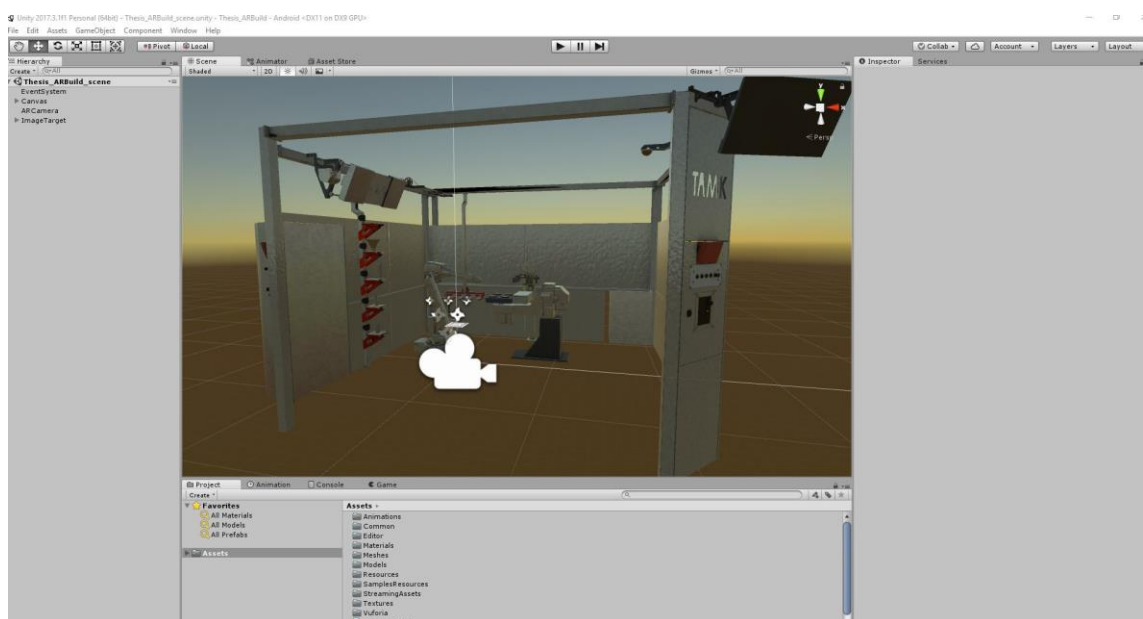
### **2.2.2 Animation for AR**

Computer generated animations are a tool to mimic or over exaggerate real-life motions. They help add realistic content to an object to invoke true realism in a computer-generated object. This realism of an objects motion draws the viewer to a more relatable state allowing the viewer to contemplate and comprehend information and data being presented to them. The ability to show what a robotic welding arm can do and its movements are key to the success of this app. Adding animation to my 3D model will be accomplished in a different 3D software, Maya. Maya is comparable to 3Dsmax but surpasses it by easily being able to create animations. My 3D models are imported into Maya and I start to create animation layers of four different movements the robotic welding arm will have. Relying on reference video I had recorded earlier of the robotics arm movement, I can reference this video for timing and movement so my animations will closely match the movement of the real robotic arm. Keyframes are created by moving the position and rotation of my 3D model then adding a keyframe or essentially a snapshot of the orientation of my model. With multiple snapshots of the model's orientation the Maya software can combine these snapshots or keyframes and create a moving animation of my model in between each keyframe. Accordingly, I can export my model as an fbx file from Maya into Unity with the animation data so the animations created will play in Unity.

### **2.2.3 Scene Creation in Unity**

The idea of this app is that it depends heavily on 3D graphics and the use of models with animations along with some elements utilizing a physics system. These requirements are created using dedicated game engines that have enhanced designs and systems that are easily used by app developers. Some of the most common 3D game engines are Unreal, Panda3D, Blender game engine and Unity. Based on the ease of use and the compatibility with AR frameworks such as Vuforia, I have decided to use Unity which is compatible with both Android and iOS platforms. Unity is a user-friendly app development editor with a powerful 3D environment platform. It is also used for 2D game development; however, the UI of the app will be the only 2D elements used.

Vuforia, the AR plugin tool, is integrated into Unity, making it easy to create Augmented Reality apps, therefore Vuforia must be activated to work in the Unity project. Under Player settings, Vuforia can be activated under the “XR Settings” section and select what platform the app will run in. The OnePlus 3 phone on loan from TAMK University runs on the Android Operating System (OS) which is chosen in the platform menu accordingly. Subsequently, activating Vuforia plugin adds all the tools and scripts Vuforia has made to work inside Unity. At this point an “ARCamera” is added to the empty Unity scene, the camera is found under the drop-down menu “GameObject”. The “ARCamera” will have the ability to lock on to the target image and render any 3D objects emerging with the live feed from the OnePlus 3 camera. With the “ARCamera” added to the Unity scene I can bring in my target image that will trigger the start of my AR scene in my app, this is also located in the drop-down menu “GameObject” under “image”. The target image will be added, parented to the “ARCamera” than the marker image is uploaded from the database made earlier in the Vuforia website. This is done by selecting the “ARCamera” in the scene and opening Vuforia configuration button under the “Inspector” menu. In this menu the app License key is inputted that was copied from the Vuforia website. With the key serial number inserted the ‘Database’ is added. Loading the correct ‘Database’ to my scene will upload my target image, this triggers the “ARCamera” to turn on whenever the marker image is viewed by the camera phone. Nevertheless, I have set up the AR operation of my scene and I can focus on adding my 3D models and animations.



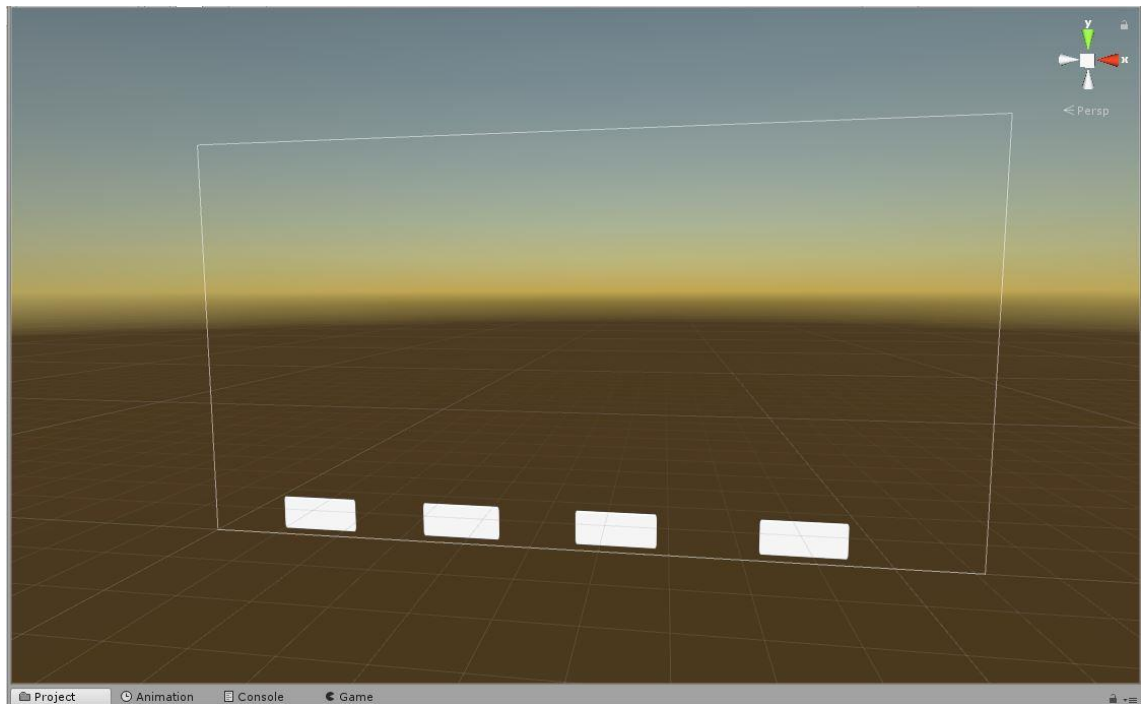
(Figure 8) Unity Integration

After importing my 3D models in to the assets folder and adding them to my scene as child objects of the target image. By parenting my models with the target object, this automatically sets up all physics and behaviours need for my app by Vuforia. Importing my models as fbx files consequently adds my animations and allows for a script to be written to play each animation separately with a push of a button, which will be created later when my GUI is built. For written script to know what animations to play at what time I separate my animations to individual clips and name each clip, as soon as each animation is named I can start writing out the script. (See Appendix 1, A)

#### **2.2.4 GUI Creation**

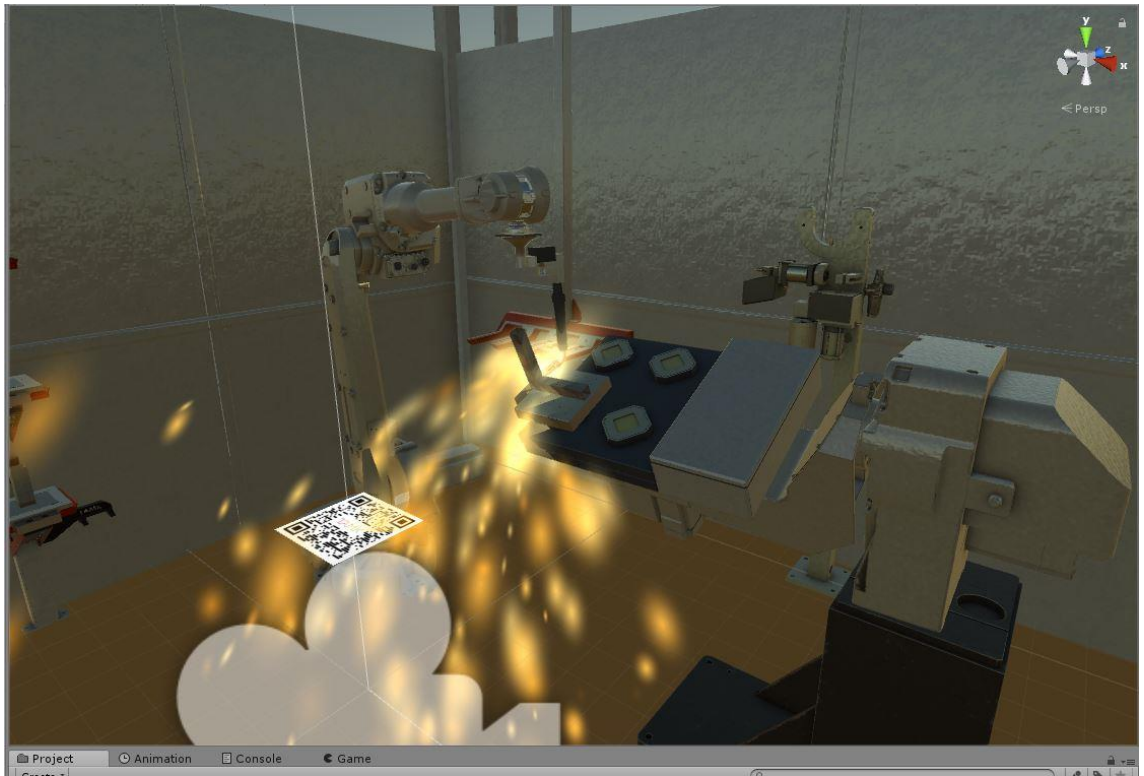
Graphical User Interface (GUI) is using graphics and images that allow the user to interact with the mobile device and the application. With the adaptation of the touch screen on mobile devices, this allows a more user-friendly interface with software. If the GUI is poorly designed it may cause the user to not want to interact with the app and not retain the informational data and be aware of the environment that is presented to them though the AR app. The design goal of the GUI is to have a simple and efficient, user-friendly interaction between the software and the user, allowing for a seamlessly output of environmental awareness and data retention.

Creating the app's GUI allows the user to interact with the AR elements by selecting or pressing a button displayed on the mobile devices screen and deciding what job the robotic welding arm will work on in their viewport and give them an idea of how the arm moves and operates. Unity has its own UI system that can display and operate on any mobile device, to enact Unity's UI a canvas must be added to the Unity scene. The canvas is the area that all User Interfaces (UI) elements will be and they will be under the canvas's hierarchy. I added four buttons to the canvas and positioned them so that they will always be at the bottom of the screen, easily assessable. Each button, when pressed has the script that starts an animation, giving the user control over which mechanics the robotic arm will operate.



(Figure 9) GUI

Wanting a more entertaining view for the user, I created a particle system attached to the welding gun. When the robotic arm is active, the viewer will see welding sparks, giving them the indication of what operation the arm is performing and giving the viewer an idea of the safety concerns needed when around this machine. This particle system will turn on with the push of an equivalent button and another small script is need for this procedure. (Appendix 1, B)



(Figure 10) Particle

### 3 Goal and Overview

So why create an app using AR? The simple answer is immersive computing. It's a strong belief that interacting with data in a more natural way is the future and AR can be beneficial by removing a lot of unneeded computing distractions between a user and data. AR utilizes a large collection of new wave technology; mobile devices, software, sensors, cameras, GPS. All accumulating to layer on a view of the real world.

This app is to help understand and establish a proof of concept allowing to visualize the mechanics and motion of a robotic arm that is operated in the engineering department of TAMK University. This AR technology can help students and engineers understand machines that generally cannot be viewed or interacted with while in operation. When a user can understand the machines, they are then able to gain first person experiences that will add with safety, operation and with this comprehension there is the ability to even help explore their process of design and function. This type of experience can allow AR applications to be created for machine maintenance, providing workers with mechanical part information from a database. Computer-generated augmented instructions can be helpful and feel like a more natural way to display assembly or operational data to a user. It can also help in assembly requirements for machine maintenance workers. An AR application can show directions to the assembler at every step of the way to building a functioning mechanism. The system can display instructions on a head mounted display for example and the user can collaborate with it by using voice commands and gestures. The goal of this app development paper was to have a starting point where AR app development could be expanded upon.

This can be beneficial for industrial and educational facilities, the usefulness of it can be easily expanded on, adding a greater variety of machines. Not much extra time is needed for adding different animation elements and 3D models. Most of the time spent working on this app was focused on the models and animations. Intergrading CAD models takes most of the time, CAD files are extremely large and if an expansion of this app was to include multiple machines, reducing those models would have to be scheduled in an appropriate method that increases efficiency in the creation and expansion of this application.

Overall, one of the main goals of creating an app of this nature is to present a starting point to allow a general industry to entertain the idea of adopting this new technology. People and industries tend to adopt new technology based on how easily it can be integrated into their lifestyle or how efficiently it works. (Technology adoption life cycle) The new technology must contribute a more of an advantage than previous methods and

it must be simplistic and observable to the user. Using mobile devices has entered a majority of people's lives and the integration of AR to the industrial environment is the next step. The main problem is getting industries to start using this type of service.

### **3.1 Problems Raised with App Creation**

In practice the AR elements jump around and are not grounded to the environment floor, they have no orientation connected to the real-world environment. This problem arises from using a target image and the scale of my 3D models. Projecting the 3D models in real world scale makes the user step away from the target image so that they can look around the AR environment. This causes the camera to lose optical tracking from the target image. Fixes for this would be to scale up my target image, doing this can be problematic regarding printing out a very large poster sized image (which I'm unable to do at this point) or painting the image on the floor of the engineering department. With this thesis of development, I don't have the permission to do such a task.

A Second option would be to use "Ground Tracking" provided by Vuforia as a tool set to use. While developing the app and adding a "Ground Tracking" to the scene would allow the app to always determine where the ground plane is oriented in the real world when the app is being used. This technology is limited to certain phone capabilities so it would limit who can view the app, also the demo phone doesn't have the needed technology to use the "Ground Tracking" option.

Some issues also were recognized during development regarding the particle system. While the particle system in the app visual adds content and a more advanced understanding of the robotic arms mechanics, a problem of the particle system not turning on consistently has been viewed. Randomly the particle system activates or not and I have been unable to identify this problematic behaviour. With my limited knowledge of coding, an outside source would be utilized in the future to determined and fix raised issues such as this.

#### **3.1.1 Improving App Development**

This app and its system of AR technology was tested during and after the development stages for seamless integration of AR models. There were some variations of discrepancies that could be improved. With future work on this app I would integrate "Ground Tracking" and hopefully Geo-location depending on the future technology available for

all devices. With Geo-location added to an app of this nature I could activate the app interaction based on the location of different machinery in correlation to the app user. When a user is within and active field area of a machine the app could bring up the related model and animations attached to it. An issue with GPS information is that the signals do not penetrate building or know the distance in height changes from level to level in a building. Once the GPS technology has been improved than this would improve visual stability with the models and a more seamless interaction for the user.

Allowing the user to simply select an animation as an introduction to a new machine and to watch the mechanics seems one-dimensional and not utilizing full potential of AR technology. Some upgrades to this application could involve allowing the user to control the AR machine with a click and drag motion with the touch screen on the mobile device. This kind of interaction could allow the user to see full limitations of a machine as well as keeping the user engaged and active on the mobile device.

Engaging users with a larger selection of machine content would be an effect utilizer for producing better quality experiences of an AR application of this nature.

Advanced technology in rendering applications could help improv the integration of virtual objects. The goal is the have the virtual objects be indistinguishable from the real objects displayed in the viewport. The issue is that rendering or generating the objects must be done in real time and computing systems in mobile devices are limited by their size and storage capabilities. New techniques and hardware is needed to accomplish this in the future.

Another expansion and improvement for the app could include infographics as part of the GUI. A user could select a section of the machine clicking on it through the touch screen of the mobile device and a graphical interface could pop-up. The graphical info would include information of the selected machine part; what it's purpose is, the model number, installation and safety instructions and any other information needed. If this this type of information is added, the usefulness this app would expand beyond just introducing new users to machines. It could become a useful tool for repair mechanics needing to order a new part with installation instructions, saving time and resources in maintaining need machines.



## 4 Conclusion

I believe the tools and procedures in this thesis can be used as an outline for developing a similar agenda for industrial use of apps utilizing AR technology. AR has limitless potential as a training and introduction tool. Planning and research aimed at the operation of an AR app for industrial use could help diminish the cost associated with training programs.

One of the most exciting progresses in AR technology is the easy and associability of AR design to the common developer. No longer is expert understanding and operation of computer systems and language needed to create and AR app. AR applications are become more commonplace and integrated in our everyday lives and this type of shift, from and advanced technology used only by military and educational facilities to a household application will seep into our work environment creating a more cost effective and cohesive tool for our everyday lives.

This application, when available in the industrial environment provides a starting point of companies or education facilities to explore AR assistance. During machine introduction to new students or employees can be difficult to explain especially of a machine is in an area that is inaccessible to people. The usefulness of an AR application of this sort allows the user to engage and experience first-hand knowledge of new machines with the trepidation of entering an unfamiliar environment.

In the forthcoming AR application development, there are important issues that must be solved in terms of stability of AR models, user interface interactions and a need to conduct user evaluations. Although, I have showed how to start with a base structure and the tools needed to generate content and create an AR application for widespread deployment of experiences and introductions to machine mechanics.

## REFERENCES

Tobias Hollerer and Dieter Schmalstieg, *Augmented Reality: Principles and Practice*, Retrieved from <https://arbook.icg.tugraz.at/Schmalstieg-2016-AWm>, pp. 5 - 10

Brooker, J., 2007. The polytechnic ghost. *Early Popular Visual Culture*, pp. 189-206.

Mark A. Livingston, Lawrence J. Rosenblum, Dennis G. Brown, Gregory S. Schmidt, Simon J. Julier, Yohan Baillot, J. Edward Swan II, Zhuming Ai, and Paul Maassel, *Military Applications of Augmented Reality*. Retrieved from <https://pdfs.semanticscholar.org/2c34/7993f71c77e8bbb6c40e5caa6ba2f6a7097f.pdf> pp. 6 – 8

Chavan, Sagar, 2016. *Augmented Reality vs. Virtual Reality: Differences and Similarities*. <https://pdfs.semanticscholar.org/7dda/32ae482e926941c872990840d654f9e761ba.pdf> pp. 2 – 3

July Rapid, 2017, Which are the best SDKs for AR App Development. Retrieved from <https://julyrapid.com/augmented-reality-sdk-which-are-the-best-sdks-for-ar-app-development/>

Jonathan Linowes, Krystain Babilinski., 2017. *Augmented Reality for Developers*, pp 74 -80

Wikipedia contributors. (2018, April 5). Technology adoption life cycle. In *Wikipedia, The Free Encyclopaedia*. Retrieved, May 10, 2018, from [https://en.wikipedia.org/w/index.php?title=Technology\\_adoption\\_life\\_cycle&oldid=834327597](https://en.wikipedia.org/w/index.php?title=Technology_adoption_life_cycle&oldid=834327597)

Wikipedia contributors. (2018, April 6). Global Positioning System. In *Wikipedia, The Free Encyclopaedia*, Retrieved, May 31, 2018, from [https://en.wikipedia.org/w/index.php?title=Global\\_Positioning\\_System&oldid=843442637](https://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=843442637)

Binzer, Judith, *Industrie 4.0 Research at German Research Institutes*, Retrieved from [https://industrie40.vdma.org/documents/4214230/5356229/I40%20Research%20at%20German%20Research%20Institutes%202016\\_2.pdf/22ad818d-bbdc-4b84-87df-afa78c251048](https://industrie40.vdma.org/documents/4214230/5356229/I40%20Research%20at%20German%20Research%20Institutes%202016_2.pdf/22ad818d-bbdc-4b84-87df-afa78c251048)

## APPENDICES

### Appendix 1. Unity Scripts

A.

```
public void AnimCandy()//just a name made up in order to find for button created
{
    Anim.Play("Candy_Animation", -1, 0f);//name of the animation
    Anim.speed = 2f;
}

public void AnimFirst()
{
    Anim.Play("_FirstWeld_Animation", -1, 0f);
    Anim.speed = 2f;
}

public void AnimSecond()
{
    Anim.Play("SecondWeld_Animation", -1, 0f);
    Anim.speed = 2f;
}

public void AnimThird()
{
    Anim.Play("ThirdWeld_Animation", -1, 0f);
    Anim.speed = 2f;
}

public void NoWeld()
{
    Anim.Play("Still", -1, 0f);
    Anim.speed = 2f;
}
}
```

B.

```
public class WeldingEngine : MonoBehaviour {  
  
    public ParticleSystem WeldingSparks;  
  
    // Use this for initialization  
    void Start () {  
    }  
    // Update is called once per frame  
    void Update () {  
    }  
    public void ToggleWeldingExhaust()  
    {  
        if (WeldingSparks.isPlaying)  
        {  
            WeldingSparks.Stop();  
        }  
        else  
        {  
            WeldingSparks.Play();  
        }  
    }  
}
```