Technical Opportunities and Challenges in Mobile Augmented Reality Prototyping

CASE: Haaga-Helia UAS Freshman application

Tuomo Muilu

Bachelor's Thesis

Degree Programme in Business Information Technology

2019

Abstract

Author Tuomo Muilu	
Degree programme Business Information Technology	
Thesis title Technical Opportunities and Challenges in Mobile Augmented Reality Prototyping. CASE: Haaga-Helia UAS Freshman application	**Number of report pages and appendix pages** 51 + 4

The purpose of the thesis is to analyse the opportunities and challenges in MAR prototyping, through the case project HH Freshman. Although technological advances have made it easier to develop Augmented Reality applications, AR application design, development, and usage is associated with some challenges, which can be broadly categorized into, technological and environmental constraints and user understanding. This thesis presents the development phases of an AR-prototype application HH Freshman and the challenges in it related to the aforementioned categories. In addition, the thesis will give some insight into how AR can be built rapidly on mobile phones. For the purpose of thesis, the author focuses on smartphones and tablets as MAR devices. The thesis follows the Design Science Guidelines for Information System Research method. The outcome of this research method is artifacts that are produced by developers during the process of solving a contextual problem. The thesis provides results and conclusions about the research topic in the case project HH Freshman. The data which is collected from unit tests and technical observations from usability tests provide valuable information for MAR prototype application development. Further research should be focused into interaction possibilities in MAR applications and new emerging tools and technologies. In conclusion, the thesis author contributes the findings about technical challenges and opportunities for HH Freshman app to develop and improve the application further.

Keywords
Mobile Augmented Reality, Prototyping, Freshman Application

Table of contents

Terms and Abbreviations

AR	Augmented Reality
VR	Virtual Reality
SDK	Software Development Kit
MAR	Mobile Augmented Reality
MR	Mixed Reality
XR	Extended Reality
UI	User Interface
UAS	University of Applied Sciences

1 Introduction

Technological devices with their hardware capability and software has multiplied significantly in recent decades. From stationary desktops we have moved to handheld computers and laptops. At the same time, devices have become smaller, and have more processing power, allowing users to have flexibility to access online resources and experience the person's environment (Höllerer & Feiner 2004, 1.). Modern tablet computers and mobile phones are examples of such technological advances.

The concept of reality has also changed. Currently, people talk about Virtual, Augmented and Mixed Reality, VR, AR, MR, respectively. These computer-generated realities are all under the umbrella term Extended Reality, XR (North of 41 2018.). As part of AR, Mobile Augmented Reality's (MAR) popularity has grown with advances in mobile phone technology (Carmigniani & al. 2011, 347.).

Although technological advances have made it easier to develop Augmented Reality applications, AR application design, development, and usage is associated with some challenges, which can be broadly categorized into technological and environmental constraints and user understanding (Craig 2013, 214-217; Dirin & Laine 2018, 12.). In addition, MAR application prototyping requires special technical expertise. In MAR application development, developers must work with a 3D environment unlike in traditional 2D application prototyping.

The purpose of the thesis is to analyse the opportunities and challenges in Mobile Augmented Reality prototyping, through the case project HH Freshman, which was started in 2018 in Haaga-Helia University of Applied Sciences. The development project utilized MAR technology and an AR prototype was made for Android systems. The application was also partially available for web-browsers. The aim of the application was to showcase the school's facilities via interaction with the environment and performing a series of gamified user tasks, such as finding all the items, unlocking new maps and playing mini-games. The users of the application were freshmen students of Haaga-Helia University of Applied Sciences. A journal paper of the concept has been published in the International Journal of Educational Technology in Higher Education. (Nguyen, Muilu, Dirin & Alamäki 2018, 5-7.)

1.1 Goals and scope

This thesis presents the development phases of an AR-prototype application HH Freshman and the challenges in it related to design, implementation and user adoption. The thesis is limited to the technical challenges and opportunities that were found. The thesis does not cover the application's user adoption, or it's use for financial purposes.

The goal of this development project is to find challenges and potential solutions in MAR prototyping through the case project HH Freshman. In addition, the thesis will give some insight into how AR can be built efficiently on mobile phones. For the purpose of thesis, the author focuses on smartphones and tablets as MAR devices.

1.2 Research questions

The questions this thesis aims to answer is listed as follows.

- What were the best suitable MAR prototyping tools that were applied for HH BITe programme's Freshman App project?

- What are the challenges and opportunities of MAR prototyping in the context of the case project in the future?

These questions will be answered based on the literature review and the results gained from the experiences of the case project.

2 Research methods

The thesis author used Design Science Guidelines for Information System Research. The Design Science research method is a problem-based process of finding and designing a suitable solution for a context-based problem in a specific environment or organization. This method was utilized on predefined requirements set in Nguyen & al. (2018). The outcome of this research method is artifacts that are produced by developers during the process of solving the contextual problem. The guidelines of the Design Science method include seven steps. The first guideline is to set the goal of the project as producing a viable artifact that can solve a problem. The second guideline is to use technological knowledge to develop solutions for a context-defined problem that comes with a certain set of requirements for the solution. The third guideline is to evaluate the usefulness, purposefulness, quality and efficiency of the designed solution using evaluation methods. The fourth guideline is to provide contributing information relating to the project topic of which the solution and the artifacts were produced. The fifth guideline is to demonstrate the implementation and application of the studied and verified methods when developing and evaluating the final solution. The sixth guideline is to understand how much of a problem has the solution solved and to what degrees there are possibilities to develop more solutions or most optimal solution in the future when certain conditions are improved. The final guideline is to include the different co-creators from both technology- and management-oriented side of the problem-based project in the development process. The final guideline allows researchers to understand the context in which the problem happens, and the solution serves. Management-oriented audiences need to know how much resources to invest and how much information to provide in supporting the production of the solution. Upon receiving the support, technology-oriented audiences will be able to construct the artifact using context-based information and requirements provided by the management. (Hevner, March, Park & Ram 2004, 82-90.)

When designing an information system or software, the artifacts include not only the software or the system itself as the result but also the documentation, models, diagrams, materials, tests materials, scripts, codes, site mapping and so on. Using the Design Science research method helps the thesis author to define the problem at the core of the project's case and then find the predefined requirements to solve the problem using technological implementation, or in this case, a mobile application. Moreover, the thesis author performed evaluation of the MAR application using the third guideline's evaluation methods such as Descriptive method to confirm the utility through literature review and constructed scenarios. Furthermore, the thesis author used the Testing method to discover functional and structural of during the implementation, Experimental method to study the usability and Analytical method to study the performance quality. (Hevner & al. 2004, 85-87.)

The chosen research methods help answer the research question “What were the best suitable MAR prototyping tools that were applied for HH BITe programme’s Freshman App project?” through literature review of the utility, as well as personal experience with development tools. Furthermore, the Experimental and Analytical methods provide insight into whether the chosen tools were right for the project and provides knowledge to the second research question: “What are the challenges and opportunities of MAR prototyping in the context the case project in the future?”. MAR is very performance heavy, therefore, requiring constant revision of the artifacts produced in the case project. The Design Science Guidelines in Information System Research provides a thorough base on which the thesis can be conducted.

3 Theoretical background

This section provides the background information of different types of simulated reality and the tools used to create AR content.

3.1 VR, AR and MAR

3.1.1 Virtual Reality

Virtual Reality (VR) is defined in Oxford Dictionary as a computer-generated simulation of a 3D-environment that can be interacted with a person using special equipment (Oxford Dictionary 2018.). The special equipment mentioned in the definition are a vision covering helmet with a screen inside and sensors or controllers attached to hands. The experience is a way to fully immerse a user to a virtual world (North of 41 2018.). Pure virtual reality would capture all the senses of a person. In a survey from 2016 and 2018, user experience along with cost were viewed as the highest obstacles in mass adoption of VR (Perkins Coie 2018, 5.).

Mobile phone use in VR requires a headset and sometimes a controller where the phone acts as the screen for the VR experience, however, currently only certain phone models can run VR software smoothly (Google 2019.). Examples of VR software are Gear VR and Daydream, developed by Samsung and Google, respectively.

3.1.2 Augmented Reality

Augmented Reality (AR) originates from virtual reality. In 1968, Ivan Sutherland created the first see-through head-mounted display that utilized cathode ray tubes to generate simple visible lines (Sutherland 1968, 759.). However, the term “augmented reality” originated in 1992 from the works of Caudell and Mizell (Arth & al. 2015, 3.). The difference between AR and VR is that VR is experienced in a virtual world and AR is overlaying virtual objects into the world through a camera. Augmented reality also has sub-categories based what type of technology is used to augment the environment. They are explained briefly below.

Marker-based AR: Marker based AR uses computer vision to recognize a certain pattern in a 2D image or 3D object. These recognizable images or objects are called markers. AR software allows developers to position an image or 3D object in relation to the marker. In figure 1 an example is shown how a computer vision software analyses an image.

Figure 1. Example of how Vuforia Engine tracks an image (PTC Inc. 2018a.)

In figure 1, we see an example of Vuforia Engine's image target recognition. The engine tracks and compares features found in the image to a known database of targets. As long as the image is in view of the camera, the object will continue to be augmented in relation to it (PTC Inc. 2018a). An example of this can be found in Figure 2.

Figure 2. A teacup augmented onto an image target of rocks (PTC Inc. 2018a.)

Markerless AR: Markerless AR is also referred to as location-based AR. Instead of markers to position an object for the user, it utilizes the device's orientation and GPS-coordinates to populate the screen. Another way to create a markerless AR experience is through Bluetooth beacons to calculate the position of the device. An example of a location-based game is Pokémon Go shown in Figures 3.1 and 3.2.

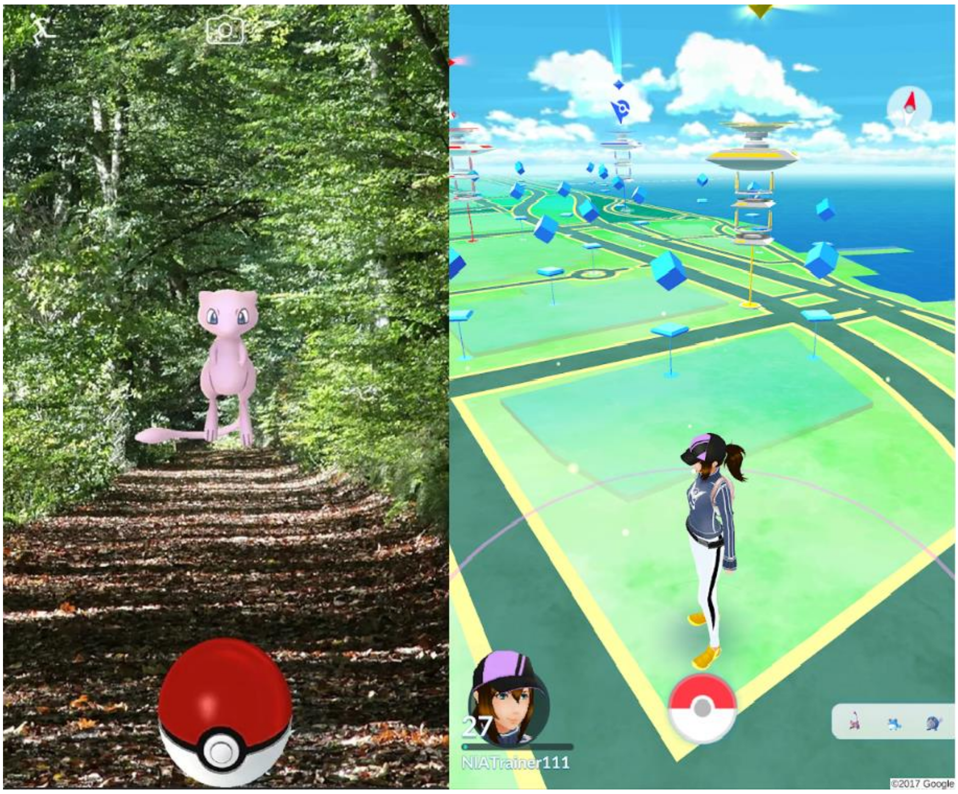


Figure 3.1 and 3.2. Screenshots of Pokémon Go from the Google Play Store (Niantic 2019.)

In Pokémon Go, the player walks around in real life and their position is updated on the map as shown in Figure 3.2. When their encounter a Pokémon in their vicinity, the player can click on them to move to the AR view as shown in Figure 3.1 where they view the world through their phone's camera and try to swipe the ball in the monster's direction to catch it.

Projection-based AR: As the name suggests, projection-based AR utilizes light to project an image to a surface and allows the projection to be interactable through sensors. An example of this is the Storyteller's Sandbox featured at the D23 expo in 2009 shown in figures 4.1 and 4.2 (Mine, Rose, Yang, van Baar & Grybdhöfer 2012, 34-36.).

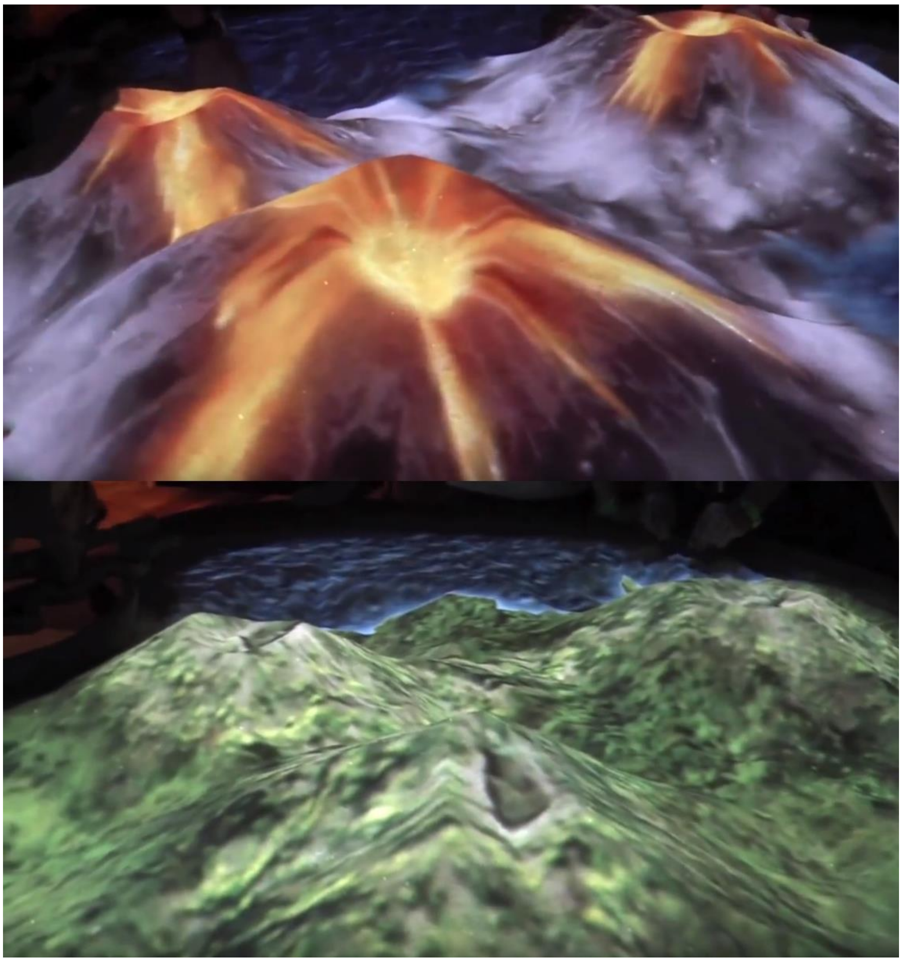


Figure 4. (4.1) Lava and mountain terrain being projected onto sand turning into (4.2) greenery after visitors plant seeds in the ground at D23 expo (Tkfore21 2009.)

As shown in figures 4.1 and 4.2, light is projected onto the sand, and sensors detect the height of the sand, changing the projection. The whole interactive experience is guided by a narrator and requires the help of visitors to shift sand in the designated areas, creating landscapes.

Superimposition AR: Superimposition AR is similar to marker-based AR. It uses computer vision to recognize an object or surrounding and superimposes an AR layer. For example, this can be used to preview furniture inside one’s home as shown in Figure 5. (Dasey 2017.)

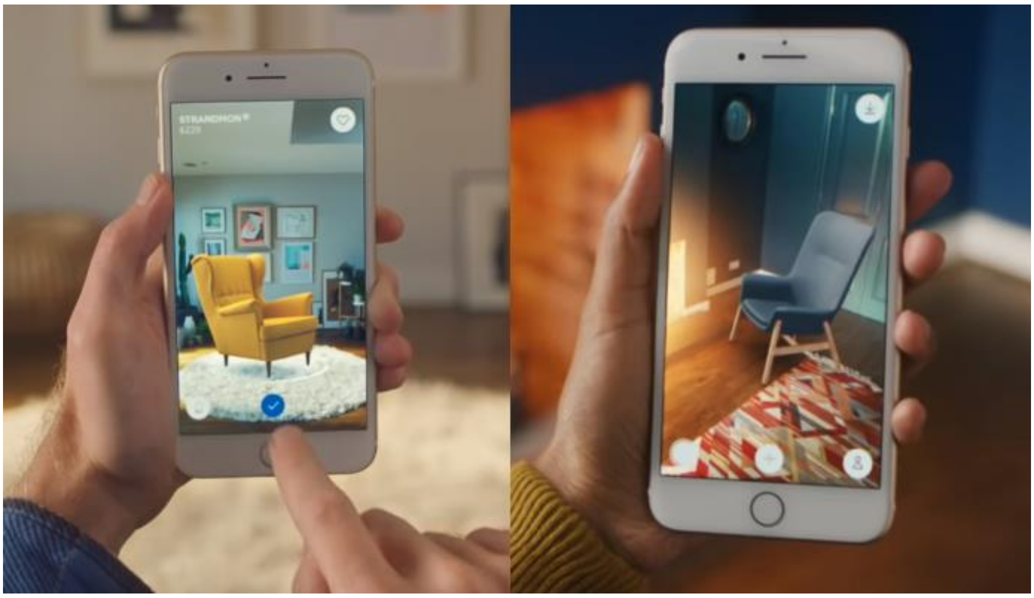


Figure 5. Demonstration of Ikea's Place application: placing true-to-scale furniture in a room (IKEA 2017.)

Ikea's Place superimposes an AR object, in this case furniture, to a room for the user to inspect. The application uses computer vision analysis to scan the depth of the room, so the application can scale objects true-to-scale.

3.1.3 Mobile augmented reality

Mobile Augmented Reality (MAR) can be any of the types of AR, marker-based, markerless, projection-based or superimposition. The term mobile refers to being able to take a device on person. On the other hand, a distinction can be made by categorizing a device as portable, if something can be moved from point A to point B, and mobile, if a device can be carried on person with ease. For example, a laptop or even desktop computer can be moved, however, operation of the device while on the move is difficult and restricted. (Craig 2013, 209-210.)

Therefore, we can state that for AR to be mobile, it must have a certain degree of ease and practicality to use. Smartphones and pad computers are mobile devices compared to alternatives such as desktops and laptops. Modern technological advances allow mobile devices to have the necessary processing power to run augmented reality. As many people already have smartphones, people do not need to buy external hardware, making smartphones the most popular devices for MAR. For this thesis, the author focuses on smartphones and pad computer to be considered MAR. Recently, augmented reality headsets or glasses are gaining traction among developers, however, a consumer product is still facing the same challenges as VR headsets do, they are expensive and require a separate purchase. (Craig 2013, 212-213.)

3.1.4 Challenges in MAR

In surveys from 2016 and 2018, user experience, for example, technological limitations and glitches were found to be the number one obstacle for mass adoption. Lack of quality content followed by cost were voted as next largest obstacles. As MAR is closely associated to smartphones, the cost is not as large a factor as in VR. Content, however, is in a pivotal role in MAR. (Perkins Coie 2018, 5.)

The challenges in creating a MAR experience can be categorized into three broad areas, technological, environmental and user understanding. Technical challenges relate to hardware and software users use and performance of the application in various conditions. Secondly, environmental challenges includes the context of use of the application and surrounding environment of the user. Thirdly, user understanding of AR consisting of user’s interaction with an AR system and how to design an application that benefits users.

3.1.4.1 Technological challenges

Although technological advances have made smartphones more powerful, there are still constraints on how detailed content phones can augment. Graphics capability in mobile phones are not yet advanced enough for realistic or high-quality real-time rendering, resulting in use of lower quality texture, or lower number of polygons in 3D models (Craig 2013, 214-215.). In MAR, content must be optimized to match the specifications of a device’s hardware. Currently, older phone models need to be taken into consideration when developing for a large audience, however, in the future, technological advances will bring powerful processors to future budget phones as well.

A key feature in the MAR experience is using the camera of the device in order to see an augmented image or object in the real world. However, depending on the quality of the camera on the device and environmental lighting, the software may have trouble recognizing a marker, or object for augmenting. This may require the user to manipulate the view of their camera unintuitively. Correspondingly, constantly using the camera and processing the image drains a device’s battery fast. Another problem a user noted in Claudia tom Dieck and Jung’s research is users may not have enough storage on their device. Developers need to be aware of different technologies with which to distribute content. Offline content that is pre-loaded in the application or cloud-based, where content is downloaded on case-to-case basis. (Claudia tom Dieck & Jung 2018, 9-12.)

Markerless AR utilizing GPS doesn't rely on the device camera for image recognition, however, the technology can be inaccurate if there's buildings or forests that block signals. The inaccuracy of GPS can lead to frustration in users. For example, in Alien Contact!, a narrative-driven GPS-based AR simulation that activates content on an approximately 30 feet radius. Certainly, GPS technology and accuracy has improved since the 2009 case of Alien Contact!, however, GPS-based applications can still face problems when accuracy needs to be within meters. (Dunleavy, Dede & Mitchell 2009, 16.)

3.1.4.2 Environmental challenges

Environmental challenges in MAR relates to the environment around the user, the context of use and how the environment interacts with the user. Whether the environment is suitable or not for AR depends on multiple factors. As mentioned in technological challenges, the lighting of the environment affects whether the camera can read a marker, however, it can also affect the user in the form for screen glare during sunny days. Mobile devices are rarely used in rainy weather and cold temperatures prevent users from comfortably using the screen. (Craig 2013, 215-217.)

Interaction in a real-time environment requires design solutions specific to the application. For instance, sounds from an application must be designed in a way for them to be heard in the environment so they do not blend or drown in background noise. The context of use of the application is important for design purposes. (Dirin 2018.)

An example on the importance of context of use is an markerless GPS-based AR application that presents an augmented bear in a thick forest. As seen from the camera, the bear will be layered on top of the trees and can break immersion as the bear appears to be floating and is lacking scale and depth to its context. Most hardware of smartphones is not advanced enough to layer a 3D object behind a real-life object and simply places the AR object in front of everything. To fix the problem, the bear could be augmented in an opening of the forest. In tourism, for example, environmental challenges such as crowds, can break immersion.

3.1.4.3 User understanding challenges

MAR is a new technology; hence, users need to learn and adapt. User understanding of AR is the interaction between user and device. Introducing a new technology poses design challenges, such as, how should a user interact with AR in an intuitive way? Should the user use their 2D screen to interact with a 3D environment, their voice or other methods?

As MAR is becoming more popular, users find themselves downloading more and more applications. Craig (2013, 217.) presents a scenario that in a hypothetical world where AR markers and content are everywhere, people point their smartphones in every direction to see if it triggers augmentable content. Currently, like non-AR applications, MAR applications typically have only one use-case. For example, IKEA Place is for placing furniture, museums have their own application for their own specific displays and AR games are meant for playing. Users must learn for which purpose an application is as each application has a different use context.

Learning in an AR system is different from traditional learning. Wu & al. (2013, 47) note that in AR learning environments, users could encounter cognitive overload from all the new information and unfamiliar technology. Hence, the design of an AR learning application needs to address how the learning process differs from learning methods used in classrooms (Wu & al. 2013, 47). On the other hand, MAR systems have provided an alternative way of learning and an opportunity to engage students with previous behavioural and academic challenges to teachers (Dunleavy & al. 2009, 18.).

Informational applications, for example, applications meant for tourism also face difficulties. Claudia tom Dieck and Jung (2018, 14) showed that there is a “link between attitude and behavioural intention as tourists with a favourable attitude are more likely to use an application than those with an unfavourable attitude”. All participants in the study liked the idea of using AR, however, some users preferred to use their own eyes to experience a place or at least limit the use of the application. One of the participants noted that the concept of tourism is to experience the destination and using an application for it defeats the purpose. This highlights the notion that AR applications need to be designed based on their context of use. (Claudia tom Dieck & Jung 2018, 14.)

As found in the case project of Barbieri & al. (2017.), users can be too focused on the screen and not be aware of surroundings. The case project details the first reported accident in medical literature of a Pokémon Go player. The pedestrian-motor vehicle collision is attributed to the inattentiveness to traffic of the pedestrian AR game player. Barbieri & al. (2017.) argue that games that involve physically moving around while looking at a device poses a significant risk of injury to users.

3.2 Tools to create MAR

3.2.1 Development Platforms

A development platform for MAR is where the mobile application is built, tested and completed. For example, in this thesis' case project, the thesis author used Unity to develop the HH Freshman application. HH Freshman is a mobile application that is built for the Android operating system. There are two major operating systems for mobiles, Android and iOS, with Android Studio and Xcode as their official development platforms respectively (Android Developers 2018; Apple 2019). Another operating system for mobiles is the universal windows platform (UWP) that is featured in Windows 10 phones.

For research purposes, the thesis author studied the official development platforms. However, Unity was chosen for the case project because of its flexibility and ability to streamline development.

3.2.2 Unity

Unity is a game engine developed by Unity Technologies that is used to create half of the world's games and over two-thirds of AR and VR experiences (Unity Technologies, 2018). Although a game engine, Unity's flexibility allows it to be used for non-game application development. Applications built with Unity can be built onto over 25 different platforms ranging from mobile, desktop, console and web-based platforms from the same C# codebase (Unity Technologies 2018.). For mobile development, Unity offers support for Android, iOS and UWP phones.

Unity also offers an in-app and browser store with assets created by community, companies and Unity Technologies. Assets are anything that can be used for a project, for example, pieces of code, user interface (UI) elements, functionality tools, or software development kits (SDK). SDK's are explained more in 3.2.5. Assets can be free or paid.

3.2.3 Android Studio

Android Studio is the official development platform for Android, developed by Google. Android Studio provides SDK's for Android versions from 2.1 (Eclair) to 8.0 (Oreo) as well as USB debugging. USB debugging allows developers to test their application on a device before implementing it. Android Studio also provides emulators to test an Android application on a PC. Unity uses the path for the SDK platforms downloaded from Android Studio to build onto Android platforms.

Applications built with Android Studio are recommended to be written in its official programming language Java, however, it is also possible to write applications for Android in C and C++ with an Android Native Kit (NDK). Android Studio applications are primarily built for Android systems, however, applications can also be deployed as web applications.

3.2.4 Xcode

Xcode is the official development platform for iOS, macOS, watchOS and tvOS, developed by Apple. Xcode is only available to macOS and cannot to be installed on Windows or Linux operating systems. Unity requires Xcode to build to iOS platforms, as it uses Xcode files found in its directory, same as Android Studio. Xcode is fully integrated with Apple developer and allows the application to be submitted straight to Apple Store.

Development in Xcode can be written in various languages. Swift, a programming language developed by Apple, is recommended due to its flexibility. Swift is a general-purpose programming language that resembles other object-oriented programming (OOP) languages like C# and Java.

3.2.5 Software Development Kits

Software development kits, occasionally called frameworks, are a collection of tools or pieces of code that allow development of applications, often through application programming interfaces (API). API’s are sets of rules that allow communication between software (Sandoval 2016.).

To understand the relationship between the technical terms better, one can imagine, that an API is a knife that is used to cut an ingredient for a meal. A SDK, however, is the whole kitchen where the process of cooking a dish happens, utilizing various tools or building blocks to complete a dish. An SDK is the framework on which to build an application and an API is a reusable function that works within the parameters of the SDK.

In the next section is a selection of SDKs used for creating augmented reality experiences. These SDKs are selected based on the experience and knowledge of the thesis author from working in MAR prototyping. The SDKs selected offer either educational or free licenses for development.

3.2.6 ARCore

ARCore is an AR SDK for Android platforms developed by Google and released in 2017. ARCore requires an Android version of 7.0 or later depending on the device and supports over 140 different device models (Google Developers 2019b). ARCore is constantly expanding the available devices for ARCore.

ARCore can also be used for AR experiences on the Chrome web browser (Medley, 2019). To some extent, ARCore can be utilized to build applications to iOS through Cloud Anchors with Unity, however, the SDK still needs an ARKit compatible device running iOS 11 or later (Google Developers 2019a; Google Developers 2019b). ARCore maps the environment through the camera and uses simultaneous localization and mapping (SLAM) technology to do so.

The tools used for developing with ARCore listed on Google Developers are Android Studio, Unity, Unreal Engine and Xcode. Xcode, however, only supports the ARCore Cloud Anchors and does not replace ARKit for iOS devices.

3.2.7 ARKit

Released in 2017, ARKit is an AR SDK for iOS devices, developed by Apple. iOS is the largest AR platform (Apple 2018b). ARKit requires iOS 11.0+ for running ARKit and the Apple Developers site provides ARKit documentation for Swift and Objective-C programming languages (Apple 2018a.). Apple released ARKit 2 in 2018 for iOS 12.0+ devices which allows AR experiences to be run on the Safari web browser (O’Hara 2018.).

Features in ARKit include the ability to use both the back- and front-facing camera for AR experiences. To create the AR experience, ARKit uses visual inertial odometry (VIO) technology for a correspondence between real and virtual spaces (Apple 2018c.). Utilizing a front-facing camera with a TrueDepth, like iPhone X possesses, ARKit can track the movement and facial expressions of a user which can be used to augment virtual masks on a user (Apple 2018a.). A popular example of this is Snapchat filters.

The thesis author does not have an Apple ID associated with a paid Apple Developer or Apple Enterprise Program to see the available tools for development and is restricted to only being able to download Xcode for an iOS device. ARKit plugins for other applications are not available to download, however, Unity and Unreal Engine do have support for ARKit on their platforms.

3.2.8 Vuforia

Vuforia Engine, acquired by PTC Inc. in 2015, is one of the most widely deployed AR platforms, supporting both Android and iOS as well as wearable technology (PTC Inc., 2018d.). Vuforia Fusion, a feature found in Vuforia Engine 7 and later can use the APIs of ARKit and ARCore on applicable devices (PTC Inc. 2018c.).

Vuforia provides powerful features such as extended tracking to maintain a reference point of an augmented object outside the view of the camera and localized occlusion detection for use in virtual buttons. Furthermore, Vuforia can recognize and track text from a set word database, which could be used for real-time translation. (Amin & Govilkar 2015, 16-17.)

The Vuforia SDK can be downloaded for development use with Android Studio, Unity, and XCode free of charge (PTC Inc. 2018d.). Alternatively, Vuforia Engine can also be used with PTC's Vuforia Studio, a web-based creation tool for the Google Chrome browser, which features drag-and-drop functionality for content (PTC Inc. 2019.). For Vuforia to run on a mobile device, an Android version of 4.4, iOS version of 11.0 or Windows 10 is needed (PTC Inc. 2018b.).

3.2.9 Wikitude

Wikitude was established in 2008 when Wikitude published Wikitude World Browser, the world's first MAR application for first Android phone, Android G1, also known as HTC Dream (Hauser 2010.). Since then, Wikitude has expanded into one of the most popular AR providers, providing support for smartphones, tablets and eyewear (Wikitude 2019a).

Wikitude's computer vision software includes three major parts. The SLAM engine, Image Rrecognition Engine and the Object Recognition Engine. Similar to Vuforia, Wikitude can use ARCore and ARKit APIs as plugins to the Wikitude SDK. The Wikitude SDK is available to Android and iOS with JavaScript and Native support and UWP with Native support. Additionally, Wikitude provides extensions for Unity, Xamarin, Titanium and Cordova. (Wikitude 2019b.)

Wikitude requires a minimum version of Android 4.4, iOS 9.0 or Windows 10 to run (Wikitude 2019c.). Wikitude also offers Wikitude Studio, a web-based creation tool with similar features to Vuforia Studio as a development platform.

3.2.10 EasyAR

EasyAR, featured in the 2016 Augmented World Expo is developed by Visionstar Information Technology and offers a completely free SDK for commercial use. Vuforia and Wikitude offer free development licences, however, to publish the application a paid licence is needed. EasyAR does offer a paid version with more features than the free counterpart. (Visionstar Information Technology (Shanghai) Co. 2018a.)

EasyAR uses SLAM technology for spatial tracking, however, it is not available in the free version. Other features in the paid version include 3D object tracking, screen recording and multi-type of target simultaneous detection and tracking. The free version, on the other hand, still provides marker-based image tracking and multi-target detection and tracking.. (Visionstar Information Technology (Shanghai) Co. 2019.)

EasyAR has the lowest operating system requirements for AR out of the selected SDKs, requiring Android 4.0, iOS 7.0 or Windows 7. As EasyAR does not generally rely on operating system APIs, it will work on newer release versions of Android and iOS. EasyAR is available for Android, iOS, Windows, macOS and Unity. (Visionstar Information Technology (Shanghai) Co. 2018b.)

3.2.11 8th Wall Web and XR

8th Wall Web is a completely web-based augmented reality platform. 8th Wall Web doesn’t require a user to download an application to experience AR. 8th Wall Web works on their own SLAM engine and is optimized to run on mobile browsers. Mobile browsers, however, need certain functionality to view AR, which translates to requirements on mobile. iOS requires a version of 11 or higher and a Safari browser, while Android requires a Chome, Chrome-variant or Firefox browser without a version requirement for the device. (8th Wall 2018a.)

8th Wall XR is an SDK for Unity to develop AR application for Android and iOS and offers the same features as Web. It uses the same SLAM engine as Web but also integrates with ARCore and ARKit APIs. 8th Wall XR requires a minimum of Android 4.4 or iOS 7.0, however, it recommends Android 7.0 or iOS 11.3 with ARCore or ARKit support. ARCore and ARKit are required for image detection and vertical surface augmentation. Devices that do not support ARCore or ARKit use 8th Wall's SLAM tracker instead. (8th Wall 2018b.)

3.3 Additional software for prototyping

This chapter is focused on providing basic information on additional 3D modeling and animation software used for development, specifically, tools that are free to use or have educational licenses.

Usually, a single software such as Unity is not enough to create a high-fidelity prototype. Because we're dealing with AR and the real-world environment, we'll need 3D models to augment. Unity has basic tools to create simple shapes that could be used to prototype, however, using an external software dedicated to 3D modeling and animations will be faster in making a high-fidelity prototype.

3D modeling and animation

In professional settings, a combination of tools is used to create models and animations. However, for prototyping, it is important for a beginner that the necessary tools from beginning to end are available for the user and streamline the development process. For example, the concept of rigging was new to the thesis author at the start of the project. Rigging is the process of adding "bones and joints" to a 3D model so an animator will know which points on the model are supposed to bend and twist. Rigging is essential for animation and can take multiple hours to days depending on the complexity of the model. Some 3D modeling and animation software streamlines this process by removing the manual labor and automatically rigging the 3D model. Below is a selected collection of 3D modelling and animation software that have free or educational licenses.

Maya

Maya is a computer animation and modeling software typically used for creating models, rigging, animation, and texturing. Further features found in Maya include working with dynamics, fluids and other simulations, lighting, shading and painting. Maya offers a 3-year educational license for students, otherwise it is paid software. (Autodesk Knowledge Network 2017.)

Houdini Apprentice

Houdini Apprentice is an educational version of Houdini FX. Houdini is a 3D modeling and FX software. Houdini Apprentice contains most features of Houdini FX and can be used for modeling, animating and rendering objects. Houdini Apprentice allows models to be exported to object (.obj) file format which is compatible with Unity as well as having a plug-in for Unity, Houdini Engine. (SideFX 2019.)

Blender

Blender is a free and open source cross-platform 3D modelling software, supporting the full 3D-pipeline. This includes modeling, rigging, animation, simulation, rendering, composition and motion tracking. Additionally, it can be used for video editing and game creation. As an open source application, users can contribute to the application by customizing it through Blender's API with Python scripting. If the community-created tools are well received by the community as good upgrades or features, it can be included in a future release version of Blender. (Blender 2018.)

Adobe Fuse CC

Adobe Fuse CC is a continuation of Fuse and found in the Adobe family. Fuse provides customizable 3D models and is available as a free beta to Creative Cloud Members. Users can change clothes, hairstyles, body proportions and details such as facial hair, eye and skin colour. Although Fuse CC has a wide range of base 3D models to start with that the user can customize, all models are humanoid figures. Fuse CC does not support creating animals or adding extra limbs to their base models. Users can upload their created character straight to Mixamo. (Adobe 2018.)

Mixamo

Mixamo is joined the Adobe family once it was acquired in 2016. Mixamo is synergetic with Fuse CC, as it allows users to automatically rig their 3D model created in Fuse CC. Mixamo also allows custom characters. Mixamo's automatic rigging web service uses machine learning to calculate the placement of the bones (Freeman 2014.). Additionally, Mixamo has thousands of different animations on their website that can be that can be applied to the model. Certain parameters of the animations can also be edited on their website, for example, jump height or distance, the speed of animation, portrayed emotion, or space between the arms. The parameters that can be changed depend on the animation.

3.4 Analysis of tools chosen for the case project

The main tools chosen for the case project are Unity, used with the Vuforia SDK with Adobe Fuse CC and Mixamo providing a 3D model and animations respectively. User interface parts were created in Adobe Photoshop and Illustrator. These tools were chosen because of their availability and adaptability for rapid prototyping.

Unity was chosen as the development platform due to its flexibility as a game engine. It's drag-and-drop visual interface was beginner friendly and there was a plethora of instructional videos made by the Unity community members regarding AR. Although Android Studio and Xcode also provided instructional videos, both tools would be needed to deploy to both platforms and they do not share the same programming language. Using both tools would have required the thesis author to learn syntax of both languages and duplicate the code. Unity provided the option to build to both Android and iOS systems from the same codebase, eliminating this problem. Xcode is only available to macOS, and the thesis author did not have a macOS device or the project budget to obtain one. Android Studio, however, with its code-focused development interface did not suit all members of the project team.

A comparative table was created by the thesis author of the SDK's featured in 3.2.2 and their features inspected whether the SDK's can support content for the different types of AR shown in 3.1.2.

Table 1. A comparison of features of SDKs and required minimum platform version

	Vuforia	Wikitude	ARKit	ARCore	EasyAR	8th Wall
Utility						
Marker-based	x	x	x	x	x	x
Markerless	x	x	x	x	x	x
Projection-based	-	-	-	-	-	-
Superimposition	x	x	x	-	x	-
Minimum platform version						
Android	4.4	4.4	-	7.0	4.0	4.4
iOS	11.0	9.0	11.0	11.0	7.0	7.0
UWP	10	10	-	-	7	-

EasyAR has the lowest requirements for a mobile device to run AR. Other SDKs such as Vuforia, Wikitude and 8th Wall require at least an Android version of 4.4 and iOS of 7.0 – 11.0. This is due to EasyAR being more focused on software calculations than device-specific APIs (Visionstar Information Technology (Shanghai) Co. 2018b.). However, the processing power of devices running the older system versions is lower than modern devices, limiting the complexity of the AR experience. Similarly, the SDKs recommend Android 7.0 and iOS 11.0 to have more reliability through ARCore and ARKit APIs.

None of the selected SDKs are marked as having the utility to produce projection-based AR, but it is because the thesis author did not find any sample use cases. However, this may be due to the SDKs being built for mobile devices. Projection-based AR requires a separate device, a light projector, to project an image to a surface. On the other hand, Vuforia, for example, allows virtual buttons that work with occlusion. Vuforia Engine could be connected to a projector-camera system to provide as a base for projection-based AR. However, current technology in mobile devices and tablets featured as the scope of the thesis have no capability to project light.

The Vuforia SDK was chosen for the case project due to its popularity as an AR SDK. Vuforia is also integrated in Unity. Additionally, compared to ARCore and ARKit, Vuforia only requires the Android OS to be a minimum of Android 4.4 and an iOS version of 9 while ARCore requires Android 7.0 and ARKit requires iOS 11. Vuforia allowed to build to both Android and iOS from Unity, lowering the time required for prototyping to both systems. Wikitude has similar features to Vuforia, however, due to the simplicity of our prototype, Vuforia was chosen due to its availability in Unity.

Personal experience was a factor in the decision of the tools. The thesis author was familiar with the Adobe Creative Suite, including Adobe Photoshop and Illustrator. Adobe Fuse CC was readily available with a subscription of the Creative Suite, therefore, an ideal choice for creating a 3D model. Maya, Houdini and Blender are all very powerful and flexible tools, however, it requires the developer to fully learn the interface and functions of the application. Without prior knowledge of the application, learning the tool for prototyping purposes takes time from other assets. Initially, Blender was used for rigging the 3D model created in Fuse CC due to prior experience with the tool, however, learning rigging for the first time was time consuming. Mixamo’s auto-rigger and pre-recorded animations were used for the final product due to its compatibility with Fuse CC models and streamlined workflow.

4 CASE: Haaga-Helia UAS Freshman application

4.1 Introduction

The case project started in October 2017 as a course work for “Digital Service Project”-course. The case project evolved into a future learning environment project and pivoted to an augmented reality orientation application for new students. The development of the application started in January 2018 and a prototype was finished in May 2018. From May 2018 till September 2018, feedback from users was collected through the application. In November 2018, a workshop was held for new students where they could test the application. The project was implemented in Unity, using Blender, Adobe Photoshop and Fuse CC, Mixamo, with the Vuforia SDK, IBM Watson SDK and Firebase SDK. An Android version and a desktop version using WEBGL were deployed, however the WEBGL does not feature an AR interaction.

4.2 Concept

The concept behind the Haaga-Helia UAS Freshman application is turning a week-long orientation for freshmen to a user-centred mobile-friendly application. Additionally, the application would act as the first introduction to the university for students coming from abroad and a way to catch up on information for those who miss the orientation week. The aim of the application is to boost the efficiency in the learning process for freshmen by providing continuous digital services and a flexible learning time at the students’ own pace. The application could also reduce staff resources and management for the first two weeks of the semester. (Nguyen & al. 2018, 5-7.)

4.3 Project framework

The case project was conducted using lean agile methodologies with the Design Science Guidelines for Information System Research. Artifacts in this case project can be considered milestones within the project such as custom QR code and functionalities of the application such as the ability to interact with the augmented 3D model.

5 Implementation

This chapter focuses on the implementation of HH Freshman in Unity and different stages of development leading to the final product.

5.1 Development process in Unity

The design of the application had two major parts. An AR scene that would have a 3D avatar welcoming the new student and then a transition to a 360-degree view, so the student could explore their new university before arriving. The application was built on Unity version 2017.3.1f1 with Vuforia 7.0.47.

5.1.1 Creating the application

The application as designed to be a mobile application, therefore the first step when creating a new Unity project was to change the build platform to Android from File > Build Settings, or Ctrl+Shift+B. The two major features were placed into different scenes to reduce memory usage during runtime. The user interface of Unity is shown in Figure 6.

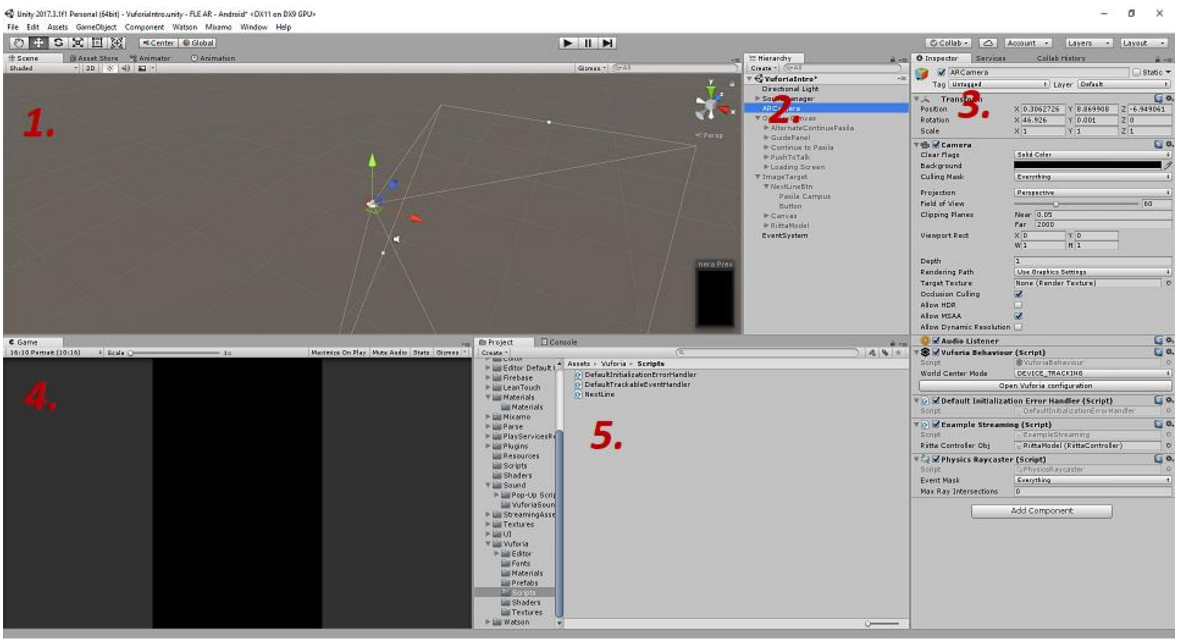


Figure 6. Unity interface with different components numbered

In Figure 6, the different components of the UI system are numbered to give a brief description of the component and its role as follows:

1. The scene viewport. It shows the content in the application and the location of the object in x-y-z coordinates. The viewport also has a tab for Asset Store, Animator and Animation.

2. Hierarchy. The hierarchy tab shows all the assets used in the scene. In this instance, the AR camera from Vuforia is highlighted and the scene viewport shows the view dimensions of the camera. Assets in the hierarchy tab can be collapsed, expanded or hidden from scene and game view from the inspector.

3. The Inspector. It is a component that shows information about the selected object, in this case, the AR camera. The inspector shows what scripts or functions are attached to it and its location on the X-Y-Z axis. Values of an object can be changed in the inspector in real-time.

4. Game viewport. The game viewport shows how the application looks from the main camera's point of view and how users would see it. In this instance, the content in the application is hidden, hence the black screen.

5. Project library. The project library shows all assets in the current project. The assets can be dragged to the scene viewport to be added into the application or from the project library through scripts. In the same section, there is a tab for Console, which shows errors in code or assets before and during runtime. Messages to the console can be sent by a Debug.log() function.

The IDE used for the case project is the default IDE installed with Unity – Microsoft Visual Studio. The user interface of Visual Studio is shown in Figure 7.

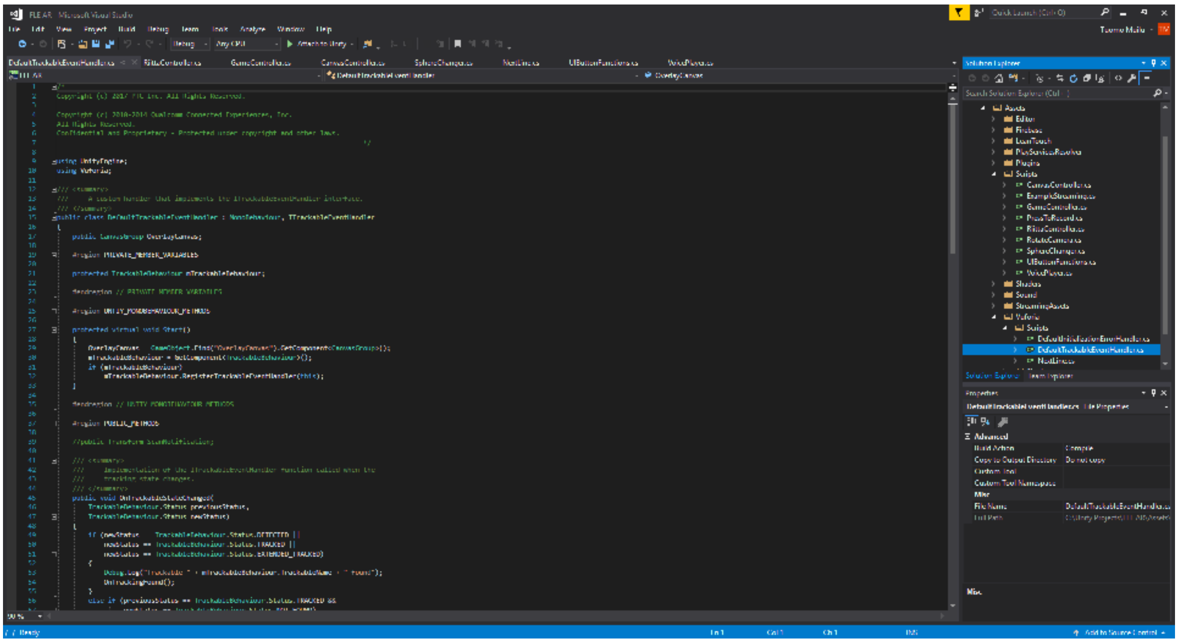


Figure 7. Microsoft Visual Studio user interface

Visual Studio acts as the main tool for writing and viewing C# scripts. To a certain extent, Visual Studio provides suggestions of functions and variable names along with support for suggesting Unity and Unity Engine specific APIs.

5.1.2 AR Scene

The basis of the AR scene is the Vuforia SDK. The Vuforia SDK was downloaded from Vuforia's webpage and added to Unity. Using Vuforia, making an object appear in AR is simple. A 3D object needs to be added as a child of the ImageTarget and placed relative to it in the Unity viewport. However, further functionality requires scripts. The first image target HH Freshman used was the back of a playing card, since it was the most readily available, without needing to print. It was also used to see how augmented reality looks as the thesis author had no prior experience with augmented reality. Vuforia Target Manager gave the image an augmentable score of 5/5. The back of the playing card is shown in Figure 8.

Figure 8: First image target used for testing: the back of a Bicycle playing card

A 3D object provided in the Vuforia SDK was added into the scene. However, since the playing card was symmetrical, sometimes the AR object would be facing the wrong direction, or suddenly flip around. A screenshot of the thesis author’s first unit test of the Vuforia Engine with a provided 3D model on a mobile phone is shown in Figure 9.

Figure 9. Screenshot of first test of augmenting an 3D object on a playing card with Vuforia

The 3D object augmented onto the playing card does not have any texture, as the thesis author was not proficient enough in 3D models in Unity to implement them at this point, despite being available in the SDK. After initial testing, the playing card was switched for an asymmetrical image to combat the model changing direction. The new image target was created in Adobe Photoshop using basic rectangle shapes and printed out in A4 and A5 size, shown in Figure 10.

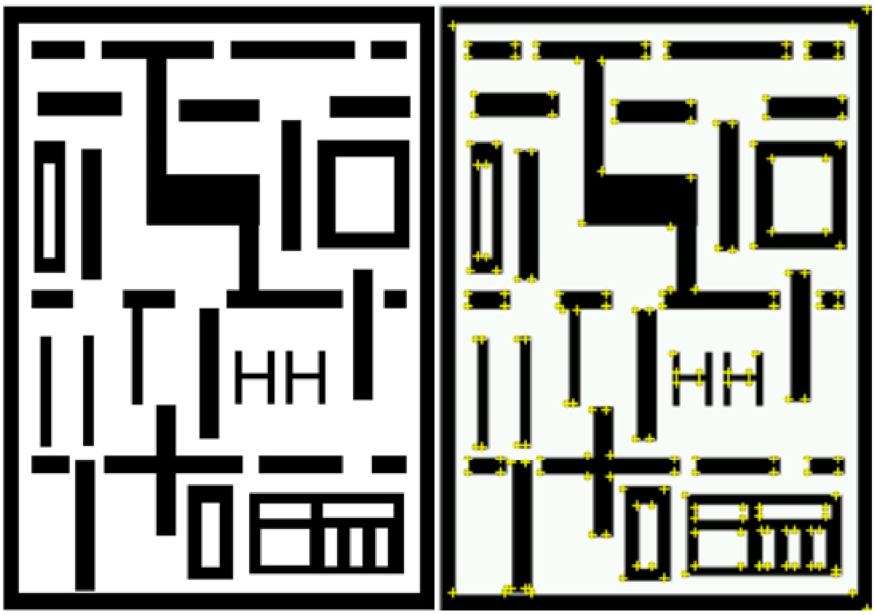


Figure 10: The first image target and the tracking features shown in Vuforia Target Manager.

Changing the image target into an asymmetrical image and more contrast between black and white made augmenting the 3D object more reliable. According to the Vuforia Target Manager, the updated asymmetrical image target had a rating of 4/5 for trackability. The 3D model used in the final application was created using Adobe Fuse CC and uploaded to Mixamo for rigging and animations. From Mixamo, the model was converted to a .fbx file and imported to Unity. Figure 11 shows the 3D model augmented on the customized image target.

Figure 11. 3D model augmented onto image target with a virtual button on the lower right

In Figure 11, the 3D model seems to have black sunglasses, however, it should have had normal see-through glasses. This was a problem with shaders in Unity, specifically with models imported from Fuse CC. There was the same problem with shaders with hair and eyelashes, as they appeared as a single block. By changing some of the default settings, the hair, glasses and eyelashes were set as created originally.

Figure 11 also has a Pasila Campus virtual button augmented onto the lower right corner of the image target. By hovering the user’s physical hand over the button in view of the camera, the button is triggered, and the application executes a piece of code in the NextLine script shown in Figure 12.

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class NextLine : MonoBehaviour, IVirtualButtonEventHandler
{
   ...

    public void OnButtonPressed(VirtualButtonBehaviour vb)
    {
        Debug.Log("Button pressed");

        //Debug.Log("This is the value of allanimplayed: " +
riittaController.allanimplayed);

        if (riittaController.allanimplayed == true)
        {
            if (callOnce == false)
            {
                StartCoroutine(LoadingScreen());
            }
        }

        else
        {
            Debug.Log("You still have to play the other animations");

            continueCanvas.SetActive(true);

        }


        //SceneManager.LoadScene("FLEAR", LoadSceneMode.Single);
    }

    public void OnButtonReleased(VirtualButtonBehaviour vb)
    {
        Debug.Log("Button released");
    }


   ...

}
```

Figure 12. Code excerpt from script handling screen transition.

The script uses UnityEngine and Vuforia collections, called in the beginning of the script. By adding Vuforia's IVirtualButtonEventHandler, the developer gains access to Vuforia's virtual button API's, namely VirtualButtonBehaviour, which automatically notices when the button is "pressed". Before the function OnButtonPressed, there is code initializing riittaController as a variable from another script, which controls the data of the 3D model, named Riitta after the teacher it was modelled after. If the button is "pressed" when Riitta has completed all her animations and voice lines, meaning that the user has completed all tasks, a Coroutine named LoadingScreen is started. LoadingScreen fades the screen and loads the next scene while showing a progress bar. If the button is pressed when the user has not completed all tasks, a pop-up is shown asking if they want to continue to the 360-degree scene despite not going through all the options.

The image target was changed for the final application. The application needed a way for users to download the application to their phones, therefore, a QR code leading to a landing page with options for a desktop or Android version was made. Additional elements were created around the QR code to give more stable tracking and a dedicated area for the virtual button on the lower right. The final version of image target has trackability score of 5/5 and is shown in Figure 13.

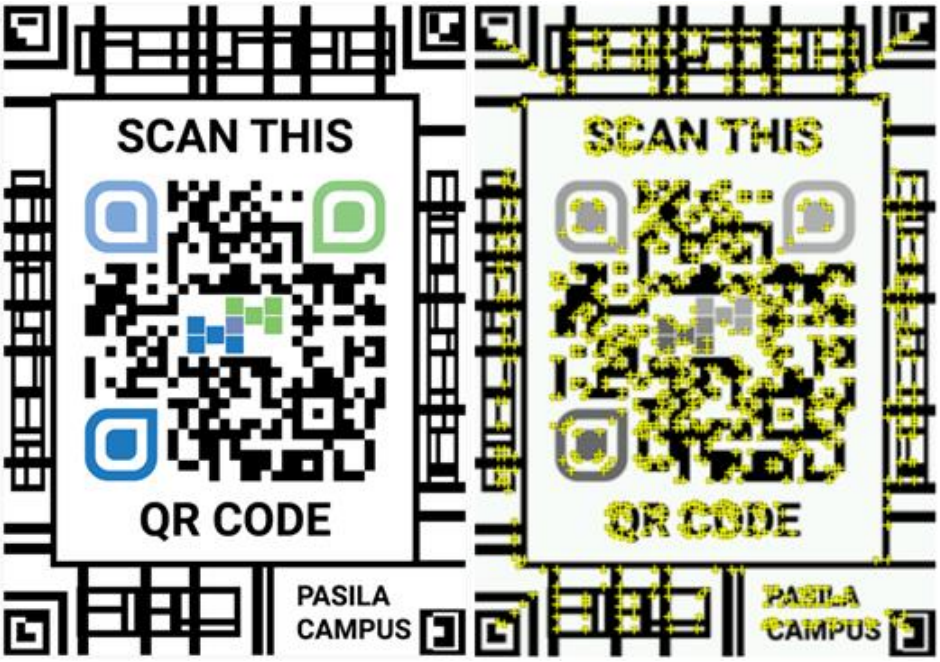


Figure 13. The final image target used with QR code and the tracking features shown in Vuforia Target Manager

The AR character was rotated 90 degrees and given a simple wall and floor to stand on. The final AR scene is shown in Figure 14.

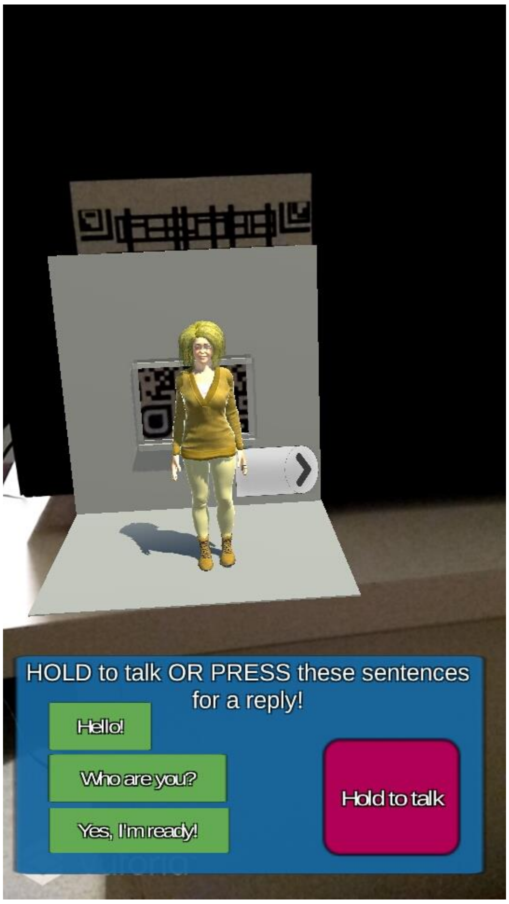


Figure 14. Final UI of the HH Freshman application's AR Scene

As shown in Figure 14, there are two ways to interact with the AR character. One way is to press the sentences, "Hello", "Who are you?" and "Yes, I'm ready" for a reply, and the other way is to hold the "Hold to talk"-button and use audio for a reply.

The voice recognition was implemented using the IBM Watson SDK. For voice recognition to work, a speech-to-text licence key is needed from IBM Cloud and added to the ExampleStreaming script. The ExampleStreaming script provided in the IBM Watson SDK also needed to be modified so that the voice is not streamed indefinitely to Watson. This was done due to possible background noise interference, making voice recognition unstable in high noise environments as well as for the 100-minute limit per month in the free Lite Plan of IBM Cloud. Additionally, the ExampleStreaming script only sends the voice-to-text script after a noticeable pause in noise levels, indicating the end of a sentence. After unit testing the voice-to-text feature, a "Hold to talk"-button was decided to be added that initialized the streaming service and ended the stream after release. A code sample from the streaming service to the 3D model is shown below.

```
...
private void OnRecognize(SpeechRecognitionEvent result)
    {
        if (result != null && result.results.Length > 0)
        {
            foreach (var res in result.results)
            {
                foreach (var alt in res.alternatives)
                {
                    //ResultsField.text is for reading the input that Watson gives,
helps developing

                    //string text = string.Format("{0} ({1}, {2:0.00})\n",
alt.transcript, res.final ? "Final" : "Interim", alt.confidence);
                    //Log.Debug("ExampleStreaming.OnRecognize()", text);
                    //ResultsField.text = text;

                    //Final result
                    if (res.final == true)
                    {

                        Debug.Log("res.final = true");

                        //Send voice command to RiittaController
                        RiittaControllerObj.RiittaActions(alt.transcript);

                    }
                }

       ...

       }
```

Figure 15: Code sample from ExampleStreaming, sending transcript to AR model

The modified ExampleStreaming script is originally part of an example showing text created from audio in real time. The script is modified by commenting out the original code and adding an if statement for when the final result, res.final, is achieved. After the final result of the transcript record is achieved, meaning there is a pause in the audio, the transcript is sent to the script controlling the AR model Riitta to the function RiittaActions. RiittaActions parses through the received transcript and based on keywords in the transcript gives an appropriate response. For now, she responds to “Hello”, “Who are you?” and “Yes”. A UML sequence diagram of the interaction is found in Appendix A and is discussed further in the functional evaluation.

5.1.3 360-degree scene

In the 360-degree scene, Vuforia is no longer needed and must be disabled to free resources on the device. By default, Vuforia will still run in the background using the camera even though the scene does not consist of an AR camera or other Vuforia resources. A Vuforia Behaviour script needs to be added to the main camera of the 360-

degree scene and the script unticked to disable Vuforia components. The scene consists of multiple spheres that each have a 360-degree panorama picture placed inside. The panorama pictures were taken around the Pasila campus with an iPhone SE using the Google Street View application. For the panorama pictures to appear on the inside of the spheres, a special shader is needed. Futhermore, panorama images need to be 2:1 width to height image ratio to stretch properly inside a sphere. The shader used in the case project is a InsideOut shader found on an instructional video (Dingle 2017.). The shader allows an image to be used as a texture on the inside of a sphere.

The 360-degree experience works by placing the main camera in the middle of the sphere and jumping from one sphere to another. Navigation between the 360-degree spheres is done by clicking on an animated ring that triggers a quick transition fade sequence to change spheres. Clicking, and other interaction such as rotation happens by adding a Lean Touch asset from Unity Asset Store. Lean Touch has a collection of scripts for enabling interaction through modern smartphone gestures such as swiping, pinching and zooming (Wilkes 2019.). Figure 16 shows the final scene finished in the Unity Editor.

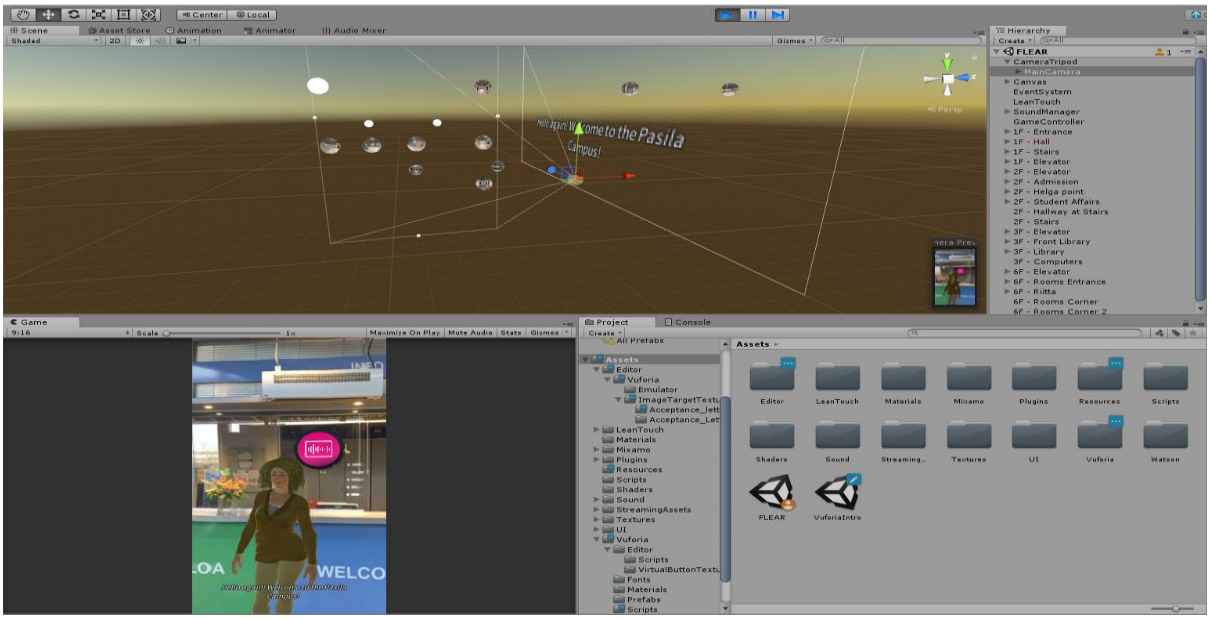


Figure 16: Final 360-degree scene in Unity Editor

The spheres shown in Figure 16 are on different levels of the y-axis to represent the different floors of Pasila. Floors 1, 2, 3 and 6 of the Pasila Campus were implemented. Since the camera retains its rotation value, the spheres were placed and rotated in the order of their real-world counterparts to avoid unnecessary labour of manually going through each sphere and coding the correct rotational values. The game view in the Unity

Editor shows a magenta-coloured speech icon. Clicking this initiates a voice line of Riitta giving useful tips about the campus in that location. When first arriving to the 360-degree scene, the user is given a quest to find all the speech icons and learn about their school.

At the end of the virtual tour on floor 6, the user is asked to rate their tour with four options:

"It was an amazing tour!"

"I'm glad that I get a chance to see the campus before I arrive."

"It was okay, I now know something I didn't know before."

"It was a hassle. I wish I didn't have to do this at all."

The data is collected to a Firebase database using a Firebase SDK for Unity. After the tour, the users are directed to go to a website to learn about the different study paths the programme has to offer. The website offered four different mini-games representing the four study paths of the Business Information Technology programme. The mini-games were created by a third party and only supported desktop browsers.

5.2 Publication to Google Play and WebGL

The final application was published to the Android system through Google Play and a browser version was built for WebGL, deployed on the personal hosting account of the thesis author on school servers. WebGL does not support Vuforia, therefore, the AR introduction had to be removed and the WebGL version only contains the 360-degree tour. iOS platforms can only be built from an macOS device, therefore an iOS application was not built as the thesis author had a Windows OS.

5.3 Data collection process

Unit tests were conducted within Unity using the console for debug messages and with the thesis author's mobile device. The mobile device is a Huawei Honor 5X. In-app reviews of the tour are stored in a Firebase database anonymously with a randomly generated string to differentiate sessions. Additionally, observations from two usability and functionality tests conducted in April and November were written down by the thesis author. The April usability and functionality tests included six participants all who were individually interviewed and recorded. The April participants were splits into two groups, three who have not attended Haaga-Helia orientation week and are not part of the school, and three that have experienced the orientation program. The November usability and functionality workshop had 19 participants in a single room testing the application. The

November participants were students who enrolled in September and a few had experienced the application during the summer. Afterwards, they filled out a questionnaire based on their experience of the application. The thesis author wrote down personal observations found during the tests.

6 Functional Evaluation

Functional evaluation of the application is conducted according to the feasibility requirements set and scenario in the final work report for the case project (Nguyen & Muilu 2018, 4-7.). The details of the scenario and requirements are included in Appendix B and C respectively. The five tasks groups shown in the affinity diagram are: Checking information, Learning the school system, Socializing, Receiving student benefits and Making their own timetable. Furthermore, unit tests for functionality of the features in the application were conducted. The results given below are based on unit testing observations, comments and personal notes from the two usability tests.

All features of the application were tested with unit tests throughout the development process. A debug log with relevant information to the execution of the function was printed to the Unity Console. Logging to console was essential for inspection of errors and seeing if every statement is reached during testing. The final implementation has been tested thoroughly for insight into potential problems and improvements for the future which are discussed further in section 7.

In accordance to the scenario, the QR code in the final image target leads the user to a landing page where they can choose to download version of the application suitable for their operating system. Overwhelmingly, participants in the workshop wanted an iOS version to test on their own device.

Checking information, which includes subtasks such as searching information on teachers, course contents and activities are not found in the application itself due to time constraints. After completion of the application, the user is directed to a web page as an extension of the application which showcases the four learning paths in the form of mini-games. Usability tests showed both potential and present students found the mini-games useful. These mini-games are mentioned in the scenario as well, however, the content of the mini-games for Asio, Winhawille and Moodle were not implemented.

Learning the school system is shown in the 360-degree environment of the application. Some usability test users did not properly identify the animated rings for navigation. This may be due to the low opacity and color of the rings that make the rings blend to the environment. However, the users in test groups found navigation to be easy after the initial recognition. The usability test users found the information in the application to be relevant for new students.

The socializing task group is marked as a necessary feature of the application. Socializing is achieved through the interaction between the AR model and the user. Usability test groups liked the 3D model based on the teacher and current students could recognize her. The “Hold to talk” feature with the 3D model caused some confusion with users. The way the button is implemented requires holding the button for the duration of the conversation.

As shown in Appendix A, pressing the button for the first time connects the user to IBM Watson through credentials and then starts streaming the audio. However, if the internet connection of the user is weak, the initial connection to Watson may take longer than for the person to initiate conversation with “Hello”, expecting a reply but it didn’t start streaming audio before they spoke. Additionally, saying “Hello” and letting go of the button before a reply does send the audio to Watson, but the voice-to-string reply is queued and is only sent to the 3D model the moment they press the button again to start streaming audio. This resulted in them repeating “Hello” and the 3D model responding in the middle of “Hello”. This caused doubt in participants and were questioning whether they did it correctly. This was a functional failure. There was no notification to the user about failure of when the application is ready to accept audio.

In the first usability test, the button was named “Push to talk”, however, it was changed to the current “Hold to talk” in an attempt to lessen confusion. Further usability tests showed that “Hold to talk” was equally intuitive as “Push to talk”. Therefore, the implementation of the feature was at fault. The intuitive implementation of voice recognition is a feature that the thesis author did not have the expertise to solve, partly due to time constraints in the project.

Furthermore, the initial phrasing from the 3D avatar regarding the virtual button in AR caused confusion. Initially, the 3D avatar stated, “All right, when you’re ready, try to press the button next to my feet!”, which caused users to press the augmented model on their phone screen. The reply was changed to “All righty! When you’re ready, use your hand to cover the Pasila Campus area on the QR code”, which was more intuitive for users interacting with the new technology. When the users hovered their hand in front of the image target and the transition started, they were delighted and interested. Regards to socialization, in the 360-degree scene, the user followed the 3D avatar around campus, listening to useful information. However, in the current state of the application, there is no interaction between students using the application, hence socialization is half-implemented.

Receiving student benefits are not in the application because the application is still in its infancy and not integrated to the school system. However, the application does provide information on where to find the benefits through the 360-degree environment, so the students know where to access the benefits at the beginning of the school year, a feature close to the scenario. User tests suggest that the information given in the application is sufficient for new students. As the application is not officially integrated into school systems, the Making their own timetable subtasks and part of scenario are not featured in the application and will be discussed further in section 7.

There were some overall problems found during testing that are not categorized in the task groups. These problems are related to the technical aspects of the application.

During usability testing, the lower part of the image target with the Pasila Campus button area was not visible as the image target was not fully scaled to the screen of the testing laptop. This led the user straight to a popup asking if they want to move to Pasila Campus on start-up, which caused confusion for the user. This happened due to Vuforia's feature of augmenting by recognizing the image target partially. The feature is meant for better and more stable tracking, however, the software assumed the virtual button was covered up which triggered the sequence. This meant the placement of the virtual button was not intuitive.

One user's smartphone in the workshop test glitched and made the application unusable upon launch. The UI of the application was fractured everywhere, and the camera was not working. This may be due to incompatibility with the Android version, however, the user could not specify which Android version their smartphone has to support this claim. Another possibility is that the smartphone had a unique resolution that was not compatible with the application.

A majority of users needed to download QR reading software to access the landing page. Reluctant to download an application just for the purpose of a test led to a few typing a direct link to their web-browser or searching for the application by name in the Google Play store.

7 Discussion

The selection of proper technology at the initial phases of a project is important. As one of the most used game engines, Unity gave flexibility and an intuitive environment for development. With the abundance of information found online in the forms of tutorials and documentation, creating with Unity for the first time was relatively easy. Compared to alternatives of Android Studio or Xcode, Unity was more beginner and user friendly.

The vision software used, Vuforia, was the correct choice of AR SDK. Vuforia provided all the necessary features required for the application, and the integration with Unity streamlined the process. Vuforia, compared to the alternatives supported the oldest Android version, 4.4, the same as Wikitude. However, Wikitude did not support a virtual button feature. EasyAR is a free alternative to Vuforia, which doesn't require a license to publish the application, however, it also does not support the virtual button. Although the virtual button caused some problems in initial tests of the prototype, the final product was worth it. For example, for a new user it was not intuitive to assume that the shapes on the sides of the image target mean anything more than visual stimuli. Without knowing the technology beforehand, and the importance of tracking points, the users were more focused on the large "SCAN THIS QR CODE"-text above and below the QR code. Users scaled the screen for the QR code for the mobile device to recognize the QR code, leaving the Pasila Campus part hidden below, triggering the pop-up. The problems encountered with the Vuforia image target can be corrected by having a square image target which would scale equally on a screen. Another possibility is rotating the image target 90 degrees for a rectangle shape that fits the common 16:9 ratio computer screens.

Future development of the application is dependent on the design choices. 8th Wall is a suitable replacement for the WEBGL browser version, which does not have AR. 8th Wall supports image tracking for ARCore and ARKit supported devices, therefore, the same image target for the case project can be used (8th Wall 2018a.). The content is already available in Unity, however, it still requires modification to transfer to 8th Wall Web technology. Alternatively, it is possible to build to iOS with Unity without a macOS device by an online virtual machine service. Such services could be used to deploy to iOS for future versions of the application in the absence of a Xcode running device. By providing both Android and iOS versions of the application, the WEBGL browser version could be taken down. ARCore and ARKit are limited to newer models of phones, therefore, it does not meet the project requirements of being available to as many students as possible.

Technological advances in the future, however, will make budget phones able to run AR, bringing MAR to a wider audience.

The 3D models created in Fuse CC were high quality and Mixamo made rigging and animation occur with a few clicks. However, the models created in Fuse CC are made for more traditional desktop applications and are not optimized to be used for MAR. The vertices count for the Riitta model used in the AR scene is close to 21000 where the recommended number of vertices for mobile games are in the range of 10000 vertices for high detail models, 5000 vertices for normal detail and 50000 vertices total for a scene (Facebook 2019.). Especially in MAR, the count should be lower than in non-MAR applications because of the added strain of augmenting. I have worked in the field of MAR prototyping for over a year since the case project concluded and have both modeled from scratch and bought models meant for MAR. None of the models used in other projects were as detailed as HH Freshman’s model. However, the AR scene in HH Freshman is quite simple. It does not require camera movement and has only one 3D character; therefore, the performance of the application was still acceptable. On older phone models, however, the framerate drop was noticeable. For future development, if more humanoid characters or details of the setting Riitta is standing on are created, the models should be more optimized for the scene.

Furthermore, if characters other than humanoid figures are to be added, Fuse CC and Mixamo are not compatible tools to do so. I recommend Blender version 2.8 as a free alternative for modeling creatures and a streamlined add-on for rigging such as Rigify. Blender 2.8 still requires expertise in modeling and animations, however, it is a free tool and brings functionality to that of high-end paid tools like Maya and Houdini without restrictions. Blender version 2.8 was not released during the development of the case project. Blender could also be used to lower the vertices count of the humanoid characters created in Fuse CC if needed.

The content of the 360-degree environment can be transferred to an augmented real-life environment of Pasila Campus by adding markers inside the school. The 3D model can be scaled to life-size and act as a virtual guide. In the future, the browser-based games can be modified to run as MAR mini-games. The current data collection to the Firebase database is very simple, however, it has the capability to be used as a tool that gathers valuable analytics and more complex feedback from users such as time taken and completion rate. The database of the application can also be integrated to Haaga-Helia UAS’s systems to allow for further content and features such as Receiving student benefits and Making their own timetable task groups that were missing from the application. An AR implementation of the two task groups, however, does not seem practical with the current data from user tests. Creating an intuitive UI for interaction with AR requires further research. Unity is still a good development platform for the project for future development as it can build to both Android and iOS as well as web.

The results gained from the case project suggest that using augmented reality in an educational context is received well by students, and they also found the application interesting and wanted to learn more about the project. (Muilu, Nguyen & Dirin 2018, 10.). The results correspond to research done by Wu & al. (2013, 47) and Dunleavy & al. (2009, 13.), showing that there is potential in the technology for educational context, however, there are still difficulties to consider. In prototyping, however, technical challenges are more prevalent, as a viable product needs to be developed on a tight schedule and compromises need to be made. This usually involves using pre-made assets and libraries that streamline the content creation process that may not be used in the final product. For example, the speech-to-text technology used was not executed properly in the application which created excitement if the AR character responded properly and confusion when there was no response. The design choice of not streaming audio constantly to IBM backfired due to the lack of knowledge on the Watson API. The implementation of the technology is at fault rather than the user.

The implementation of the speech-to-text functionality shown in Appendix A has no feedback to the user on what is happening. The lack of feedback from the application in critical points such as authentication and giving voice commands to the AR avatar is a flaw in the design. Due to the application still being in prototyping phase, there was the assumption that authenticating credentials would always work and users would successfully communicate with the AR avatar. There is no error or notification to the user that the connection is made or that the voice command went through to wait for a reply. Additionally, the Watson SDK provides the possibility to send an audio clip to the servers, however, I was not able to make it function. A possibility is to change the Watson speech-to-text SDK to a local dictionary database that does not require internet connection to fasten the avatar’s replies. There are free speech-to-text assets specifically for Android, however, iOS would require a different SDK, or a paid SDK that offers both Android and iOS functionality.

The tools chosen for the project, Unity, Vuforia SDK, Fuse CC and Mixamo with the IBM Watson SDK were correct, however, I did not utilize the full potential of the used tools. This answers the first research question. The project may have been too ambitious and implemented too many technologies at once. A QR code for downloading the application, AR and interaction with voice recognition as well as buttons, 360-degree environment to explore and browser-based games to learn about courses. The application seems fragmented to a new user. Even though the AR technology is impressive to users, if the implementation is done badly the experience is a negative one. Based on experience gained throughout the project, the application should use the same interaction techniques used in traditional software development and apply it to new technologies such as AR to ease learning. To answer the second research question of the challenges and opportunities in MAR protoyping, a compilation of challenges encountered by the thesis author during the case project is found in Table 2 below.

Table 2. Challenges in MAR prototyping found in the case project

Challenges	Description
Technological	Making changes to pre-made assets imported due to time limitations without proper investment of time to learn the intricacies of how it works can lead to a negative experience. For example, the case with the speech-to-text API. It is hard to test the application for the variety of devices users have. For example, the assets used for traditional applications may be too heavy for MAR applications, lowering the framerate in older phone hardware. Technical limitations were rated as the highest obstacle in mass adoption of AR (Perkins Coie 2018, 5.). There are many tools for development available that offer similar functionalities, however, selecting which tool is suitable for prototyping functionality may not scale for the full application or requires knowledge of the tool beforehand. For example, the discovery that Vuforia does not support AR in WEBGL was only found after completing the mobile version. HH Freshman application required users to use too many different technologies from beginning to end. The texture sizes in the 360-degree view were scaled down to fit the full application size requirements of Google Play. Similarly, a user may not have room for the application on their device (Claudia tom Dieck & Jung 2018, 12).

Environmental	The environment during testing was a controlled space, however, the design of the application's context of use needs to reflect the variety of use cases of the application (Dirin 2018.). Participants often preferred to click the sentences than use voice recognition features, therefore not engaging with the AR. In the November workshop, participants were seated close to one another, which may have been a factor as the application was designed to be used in quiet or private.
User Understanding	Directions given in the application were not intuitive enough for users. This led to users being stuck or asking assistance. Interaction in MAR is different from traditional applications, which may have confused the users. Users were required to download and utilize too many applications for a one-time use. The fragmentation of the different features of the application to different technologies made the application too complex to use as a first-timer. Giving the option to interact with buttons or voice needlessly complicated the application.

Even though there are a lot of challenges presented during the development of the MAR prototype, there are great opportunities with MAR technology. A compilation of the opportunities in MAR prototyping the thesis author found is in Table 3 below.

Table 3. Opportunities in MAR prototyping found in the case project

Opportunities	**Description**
Technological	New technology such as 8th Wall shows promise to be able to distribute the application on browser platform without requiring a user to download external software. This eliminates some hurdles, which would otherwise stop a user from downloading an application for one-time uses. Users found the 3D avatar representation of the real-life teacher interesting. Further development to improve upon interaction technologies such as speech-to-text responsiveness and

	animations will make HH Freshman 3D avatar even more life-like. Virtual buttons in MAR bring a spatial element to user interaction, adding value for the user. Furthermore, the browser games at the end of the tour can be turned into MAR to visualise better the different study paths. As technology progresses further, more tools and practical applications for MAR are created and the capability of mobile devices to run MAR improves.
Environmental	The content from the 360-degree environment in the application can be transferred to the real world Pasila Campus. This way, the users can experience the orientation at home as well as in Pasila Campus. The context of use of an interactive MAR avatar can expand to other industries that require orientation, training or informational help.
User Understanding	MAR provides unique opportunities for users to engage with virtual objects. As technology progresses, MAR user interfaces need to evolve and provide more interaction than pressing a device screen. For example, with virtual buttons and voice recognition. Clearing up the interaction between user and MAR, could create more positive engagement and interest in the topic (Wu & al. 2013, 47.; Dunleavy & al. 2009, 13.).

Although current challenges in MAR prototyping are heavily technology-focused, resolving the technology problems provides opportunities to advance user experience. By utilizing new technologies, developers can experiment with user interaction gradually until MAR applications become more standardized to new interaction elements where a new generation of users find them intuitive.

8 Conclusion

The implementation of the case project followed Design Science Guidelines for Information System Research which provided relevant steps on how to research software development. The thesis author was involved in all technological steps of the application and hence has specific knowledge on the details of the case project. Challenges found from the case project correspond to other challenges present in AR projects. Management of the project was divided between the thesis author and a co-developer and was mentored and guided by their teacher and supervisor, Amir Dirin. The case project was the first mobile software development project the thesis author had undertaken. The co-developer is writing a thesis and covering the user experience, usability and design of the case project.

The tools selected for the case project, Unity, Vuforia SDK, IBM Watson SDK, Adobe Fuse CC and Mixamo were all great for the purpose of developing a MAR application. Unity has a lot of documentation and tutorials available online for free. Slight modification in pre-made assets fits the purpose rapidly developing a prototype. Vuforia offered the most features out of rival vision software. WEBGL not being supported by Vuforia was a setback, however, the concept was for meant for mobile devices and Vuforia had all necessary features for MAR. The IBM Watson SDK for Unity was the correct tool for implementing voice recognition. The implementation of the voice recognition feature was lacking due to the experience of the thesis author. Adobe Fuse CC and Mixamo saved a lot of time with their features.

Data collected from unit tests and technical observations from usability tests provide valuable information for MAR prototype application development. HH Freshman still has development iterations to go through before it can be applied to whole of Haaga-Helia UAS. However, the initial findings support that the application is feasible and will generate interest towards Haaga-Helia UAS.

Further research should be focused into interaction possibilities in MAR applications and new emerging tools and technologies. In general, content for MAR applications need to be optimized to have a smooth AR experience across as many devices as possible. Even though the technology is impressive, without engaging and purposeful content, the application can be considered a quick novelty. For specifically HH Freshman, the UI needs to more polished and the “Hold to talk”-button functionality more responsive.

References

8th Wall 2018a. 8th Wall Web Documentation. Available at: https://docs.8thwall.com/web/. Accessed: 23 March 2019.

8th Wall 2018b. 8th Wall XR Documentation. Available at: https://docs.8thwall.com/xr. Accessed: 23 March 2019.

Adobe 2018. Fuse CC (Beta) Common questions. Available at: https://helpx.adobe.com/beta/fuse/faq.html. Accessed: 11 March 2019.

Amin, D. & Govilkar, S. 2015. Comparative Study of Augmented Reality Sdk's, International Journal on Computational Science & Applications, 5(1), pp. 11–26. doi: 10.5121/ijcsa.2015.5102.

Android Developers 2018. Meet Android Studio | Android Developers. Available at: https://developer.android.com/studio/intro/. Accessed: 9 March 2019.

Apple 2018a. ARKit: Apple Developer Documentation. Available at: https://developer.apple.com/documentation/arkit (Accessed: 20 March 2019).

Apple 2018b. ARKit - Apple Developer. Available at: https://developer.apple.com/arkit/. Accessed: 20 March 2019.

Apple 2018c. Understanding World Tracking in ARKit | Apple Developer Documentation. Available at: https://developer.apple.com/documentation/arkit/understanding_world_tracking_in_arkit. Accessed: 20 March 2019.

Apple 2019. Welcome to Xcode - Xcode Help. Available at: https://help.apple.com/xcode/mac/current/#/devc8c2a6be1. Accessed: 9 March 2019.

Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A. & Wagner, D. 2015. The History of Mobile Augmented Reality Developments in Mobile AR over the last almost 50 years. arXiv. Available at: http://arxiv.org/abs/1505.01319. Accessed: 20 March 2019.

Autodesk Knowledge Network 2017. Basics. Available at: https://knowledge.autodesk.com/support/maya/getting-started/caas/CloudHelp/cloudhelp/2017/ENU/Maya/files/GUID-6B531DDB-3440-4216-

A322-FB6CD1EA83A1-htm.html. Accessed: 11 March 2019.

Barbieri, S., Vettore, G., Pietrantonio, V., Snenghi, R., Tredese, A., Bergamini, M., Previato, S., Stefanati, A., Gaudio, R. & Feltracco, P. 2017. ‘Pedestrian inattention blindness while playing Pokémon Go as an emerging health-risk behavior: A case report’, Journal of Medical Internet Research, 19(4). doi: 10.2196/jmir.6596.

Blender 2018. About blender. Available at: https://www.blender.org/about/. Accessed: 11 March 2019.

Carmigniani, J. et al. 2011. Augmented reality technologies, systems and applications. Multimedia Tools and Applications. doi: 10.1007/s11042-010-0660-6.

Claudia tom Dieck, M. & Jung, T. 2018. A theoretical model of mobile augmented reality acceptance in urban heritage tourism. Current Issues in Tourism, 21(2), pp. 154–174. doi: 10.1080/13683500.2015.1070801.

Craig, A. B. 2013. Understanding Augmented Reality: Concepts and Applications. Elsevier Inc. doi: https://doi.org/10.1016/C2011-0-07249-6.

Dasey, D. 2017. IKEA Place augmented reality app. Available at: https://highlights.ikea.com/2017/ikea-place/. Accessed: 7 March 2019.

Dingle, N. 2017. [Tutorial] Unity3d - 360 Virtual Tour. Available at: https://www.youtube.com/watch?v=2q9wc9Y35wg. Accessed: 20 March 2019.

Dirin, A. 2018. The challenges and opportunities of user experience in AR based applications. Available at: https://esignals.haaga-helia.fi/2018/03/01/the-challenges-and-opportunities-of-user-experience-in-ar-based-applications/. Accessed: 7 March 2019.

Dirin, A. & Laine, T. 2018. User Experience in Mobile Augmented Reality: Emotions, Challenges, Opportunities and Best Practices. Computers, 7(2), p. 18. doi: 10.3390/computers7020033.

Dunleavy, M., Dede, C. & Mitchell, R. 2009. Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. Journal of Science Education and Technology, 18(1), pp. 7–22. doi: 10.1007/s10956-008-9119-1.

Facebook 2019. Technical Guidelines and Optimization in Spark AR Studio. Available at:

https://developers.facebook.com/docs/ar-studio/technical-guidelines/. Accessed: 15 April 2019.

Freeman, W. 2014. Digging into rigging: Mixamo's character service explored. Available at: https://www.mcvuk.com/development/digging-into-rigging-mixamos-character-service-explored. Accessed: 11 March 2019.

Google 2019. Daydream. Available at: https://vr.google.com/daydream/. Accessed: 11 March 2019.

Google Developers 2019a. Quickstart for Cloud Anchors in iOS | ARCore | Google Developers. Available at: https://developers.google.com/ar/develop/ios/cloud-anchors-quickstart-ios. Accessed: 18 March 2019.

Google Developers 2019b. Supported Devices | ARCore | Google Developers. Available at: https://developers.google.com/ar/discover/supported-devices. Accessed: 18 March 2019.

Hauser, A. 2010. Wikitude World Browser. Available at: https://www.wikitude.com/wikitude-world-browser-augmented-reality/. Accessed: 22 March 2019.

Hevner, A. R., March, S., Park, J. & Ram, S. 2004. Design Science Research in Information Systems. MIS Quarterly. doi: 10.1007/978-1-4419-5653-8.

Höllerer, T. & Feiner, S. K. 2004. Mobile Augmented Reality. Telegeoinformatics. doi: 10.1201/b12395.

IKEA 2017. Say Hej to IKEA Place. Available at: https://www.youtube.com/watch?v=UudV1VdFtuQ. Accessed: 10 March 2019.

Medley, J. 2019. Augmented reality for the web. Available at: https://developers.google.com/web/updates/2018/06/ar-for-the-web. Accessed: 18 March 2019.

Mine, M., Rose, D. & Yang, B. 2012. Reality in Disney Theme Parks. Computer, 45(7), pp. 32–40. doi: 10.1109/MC.2012.154.

Muilu, T., Nguyen, N. & Dirin, A. 2018. Rapid Mobile Augmented Reality Prototyping With

Unity, Vuforia, and 3D Modeling. ICERI2018 Proceedings, 1(11), pp. 10023–10032. doi: 10.21125/iceri.2018.0873.

Nguyen, N., Muilu, T., Dirin, A. & Alamäki, A. 2018. An interactive and augmented learning concept for orientation week in higher education. International Journal of Educational Technology in Higher Education. doi: 10.1186/s41239-018-0118-x.

Nguyen, N. & Muilu, T. 2018. Final Report, Freshman's Orientation App for Haaga-Helia.

Niantic Inc. 2019. Pokémon GO. Available at: https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&hl=en. Accssed 7 March 2019.

North of 41 2018. What really is the difference between AR / MR / VR / XR ?. Available at: https://medium.com/@northof41/what-really-is-the-difference-between-ar-mr-vr-xr-35bed1da1a4e. Accessed: 7 March 2019.

O'Hara, A. 2018. AR Quick Look brings augmented reality to Safari in iOS 12. Available at: https://appleinsider.com/articles/18/07/13/ar-quick-look-brings-augmented-reality-to-safari-in-ios-12. Accessed: 20 March 2019.

Oxford Dictionary 2018. Definition of virtual reality in English by Oxford Dictionaries. doi: 10.1093/acref/9780199571123.001.0001. Accessed 7 March 2019.

Perkins Coie 2018. Augmented and Virtual Reality Survey Report. Perkins Coie, pp. 1–20.

PTC Inc. 2018a. Image Targets. Available at: https://library.vuforia.com/articles/Training/Image-Target-Guide. Accessed: 27 March 2019.

PTC Inc. 2018b. Vuforia Engine Supported Versions. Available at: https://library.vuforia.com/articles/Solution/Vuforia-Supported-Versions. Accessed: 22 March 2019.

PTC Inc. 2018c. Vuforia Fusion. Available at: https://library.vuforia.com/content/vuforia-library/en/articles/Training/vuforia-fusion-article.html. Accessed: 22 March 2019.

PTC Inc. 2018d. Vuforia Overview. Available at: https://library.vuforia.com/content/vuforia-library/en/getting-started/overview.html. Accessed: 22 March 2019.

PTC Inc. 2019. Augmented Reality for the Industrial Enterprise. Available at: https://www.ptc.com/en/products/augmented-reality/vuforia-studio. Accessed: 22 March 2019.

Sandoval, K. 2016. What is the difference between an API and an SDK?. Available at: https://nordicapis.com/what-is-the-difference-between-an-api-and-an-sdk/. Accessed: 9 March 2019.

SideFX 2019. Houdini Engine for Unity. Available at: https://www.sidefx.com/docs/unity/. Accessed: 11 March 2019.

Sutherland, I. E. 1968. A head-mounted three dimensional display. Fall Joint Computer Conference, pp. 757–764. doi: 10.1145/1476589.1476686.

Tkfore21 2009. D23 Expo: Storyteller's Sandbox Demo. Available at: https://www.youtube.com/watch?v=IypiP5KQfXA. Accessed: 22 March 2019.

Unity Technologies 2018. Unity. Available at: https://unity3d.com/unity. Accessed: 22 March 2019.

Visionstar Information Technology (Shanghai) Co., Ltd. 2018a. Getting Started with EasyAR SDK. Available at: https://www.easyar.com/doc/EasyAR SDK/Getting Started/Getting-Started-with-EasyAR.html#what-is-easyar-sdk. Accessed: 22 March 2019.

Visionstar Information Technology (Shanghai) Co., Ltd. 2018b. Platform Requirements. Available at: https://www.easyar.com/doc/EasyAR%2520SDK/Getting%2520Started/2.0/Platform-Requirements.html. Accessed: 23 March 2019.

Visionstar Information Technology (Shanghai) Co., Ltd. 2019. EasyAR SDK. Available at: https://www.easyar.com/view/sdk.html. Accessed: 23 March 2019.

Wikitude 2019a. About Wikitude. Available at: https://www.wikitude.com/about/. Accessed: 22 March 2019.

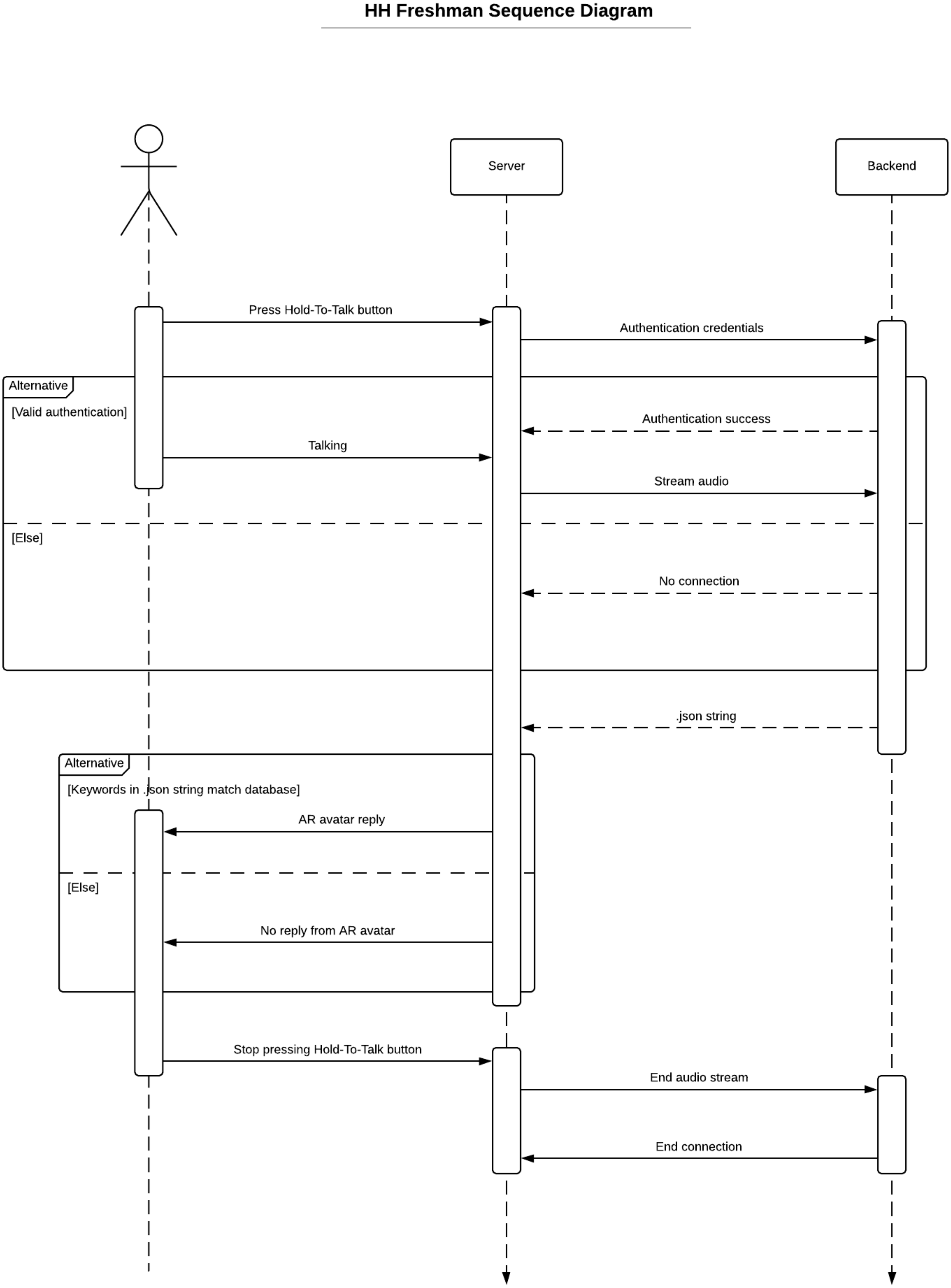
Wikitude 2019b. Getting started Wikitude SDK Unity. Available at: https://www.wikitude.com/external/doc/documentation/latest/unity/. Accessed: 22 March 2019.

Wikitude 2019c. Supported Devices. Available at: https://www.wikitude.com/documentation/latest/android/supporteddevices.html. Accessed: 22 March 2019.

Wilkes, C. 2019. Lean Touch. Available at: https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111. Accessed 22 March 2019.

Wu, H.-K., Lee, S., Chang, H. & Liang J.-C. 2013. Current status, opportunities and challenges of augmented reality in education. Computers and Education. Elsevier Ltd, 62, pp. 41–49. doi: 10.1016/j.compedu.2012.10.024.

Appendix A. UML Sequence Diagram



APPENDIX B. Full Scenario

Eric is a new student who received his acceptance letter at HH in June. He is a freshman student in the BITe programme. With the letter of acceptance, Eric also received a barcode or link labeled as **HH Orientation Game**. Eric accesses the link and loads the game on his mobile phone. He can then start to **learn** about HH and BITe while playing the game.

Within a few days, Eric has gained **visualized knowledge** about his campus (Pasila), his HH profile and teachers as well as **practical information** of using campus and online services. By playing the Orientation Game featuring a **virtual tour**, Eric now knows which services are located where in the Pasila campus such as the library, computers and printers, cafeteria services and so on.

He is able to use **school benefits** such as M-Drive, VDI, Office365, and student discounts. He has also learnt the process of looking for course information and timetables, registering to courses, creating own timetables and tracking performance by playing MyNet, Asio, WinhaWille related mini-games.

By the end of the week, Eric has completed all mini games within HH Orientation that **help schedule** his courses for the first and second period of his first semester. He knows how to get the services using his school items such as library card, student credentials, and lunch card.

When the school's academic year starts, Eric goes to school and he is able to easily access and **use school services** right away. He knows what his scheduled courses offer because he has learnt about the courses' introduction through Moodle mini-games. He knows how to optimize the HH learning system such as VDI and Moodle to study more effectively without wasting time.

Appendix C. Affinity Diagram

Task group	Checking info	Learning school system	Socializing	Receiving student benefits	Making own timetable
Subtasks	Checking courses on MyNet, teachers, courses content and activities	Learning how to use Asio, Winhawille, Moodle, VDI and Pasila campus's facility	Meeting and communicating with other students and teachers	Receiving student's benefits, such as travel, lunch discount, software packages, health and fitness services	Check the academic year's timetable, choose courses, make own timetable
Task generality	General	General	General	General	General
Significance	Main	Main	Necessary	Main	Main
Frequency of use	Always	At the orientation weeks	Often	Always	Every study period
Tools	Phones, PC	Phone, PC	Phone, PC	Phone, PC	PC
Data	Courses content, requirement, materials, teachers, study level, course code, name, general info	Student profiles, student number, email, password, student cards, printer ID, paper balance, library	dashboard, email, events info, meeting info	Student profile, student number, password, email, ID	Student number, password, course code, course name, study time, classroom, course type, teachers, campus location, language, credits
Skills required	Browser and smartphone user	Browser and smartphone user	Smartphone app user	Browser and smartphone user	Browser user
User expectations	To know the most important info about potential study courses	To study more efficiently, plan study personally and keep track of study progress	To create good relationship and teamwork and get support	To support student life in school and in person	To plan personal study plan with suitable courses and to create own timetable
Team task aspects	Can always share the courses info or see how many likes the	Can learn with other (If technology is allowed in interactive virtual	Always done with others	Can always invite other students to use some benefits together	Can see how many friends or classmates that choose the same courses in own

	courses	reality)			timetable
End-user validation	Users successfully check all info needed without problems	User able to study the knowledge and know how to use it practically	User able to develop social and soft skills and connect with others of same goals	User able to earn rewards from study and works and use them to support their life	User able to create suitable timetable to their interests and preferences or in the same study groups that they prefer