

Tuomo Kinnunen

POSSIBILITIES OF THE MOBILE VR PLATFORM IN RELATION TO THE PC VR

POSSIBILITIES OF THE MOBILE VR PLATFORM IN RELATION TO THE PC VR

Tuomo Kinnunen
Bachelor's Thesis
Autumn 2018
Information technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology

Author: Tuomo Kinnunen

Title of thesis: Possibilities of The Mobile VR Platform in Relation to the PC VR

Supervisors: Pekka Alaluukas, Iikka Finning

Term and year when the thesis was submitted: Spring 2019 Number of pages: 38 + 1

The main target of the thesis was to study mobile platform possibilities, limitations, pros and cons in relation to the PC version of an VR application and to give an objective description of those in short and at least in a two-year time span. This thesis was commissioned by Nokia.

The Virtual Reality PC application already existed, and the study had to be focused on mobile VR possibilities in relation to it. The author had no previous knowledge of Unity or implementing Virtual Reality applications.

The thesis work started and continued in short iterations of study and practical work. Short iterations were implemented by first studying desired feature and then studying how to implement it in practice.

As a result, three different kinds of mobile applications were developed. Those applications were done because they demonstrated mobile platform performance in practice. The main conclusion was that the mobile VR platform is not yet ready to be used in the company's projects, but a regular touch screen mobile application could have the potential.

Further developing should focus on a mobile touch screen application in a mobile developing area. There has to be performance improvements in the mobile VR platform before it is truly usable and meets the needs.

Keywords: Android, Virtual Reality, Mobile

PREFACE

I want to thank my supervisors, Pekka Alaluukas from OUAS and Iikka Finning from Nokia who supervised me through this thesis work. I also want to thank Nokia as a company for giving me an opportunity to be a thesis worker.

Oulu,

21.1.2019

Tuomo Kinnunen

TABLE OF CONTENTS

ABSTRACT.....	3
PREFACE.....	4
TABLE OF CONTENTS.....	5
VOCABULARY	7
1 INTRODUCTION	8
2 WHAT IS VIRTUAL REALITY?.....	10
2.1 History of VR	10
2.2 VR in gaming.....	11
2.2.1 Game consoles.....	11
2.2.2 PC VR Gaming	12
2.2.3 Mobile VR Gaming.....	14
3 MOBILE VR.....	15
3.1 History.....	15
3.2 System requirements	16
3.3 Controllers.....	18
3.3.1 Supported controllers.....	18
3.3.2 Other possible controlling methods.....	19
3.4 Multiplayer in mobile VR.....	21
3.4.1 Existing application examples	21
3.4.2 Multiplayer player amount.....	21
3.4.3 Hardware requirements.....	22
3.4.4 Software requirements.....	22
3.5 Hardware performance comparison between PC and mobile devices	23
3.5.1 GPU comparison.....	23
3.6 Porting Unity PC application to mobile platforms.....	24
3.6.1 Porting to Android platform	25
3.6.2 Porting to iOS platform.....	26
4 PRACTICAL PART	27
4.1 Mobile Virtual Reality application.....	27
4.2 Flat screen mobile application	30
4.3 Mobile and pc platform pros and cons in Virtual Reality	32

5	RESULTS	34
6	CONCLUSION.....	35
	REFERENCES	36
	APPENDICES.....	39

VOCABULARY

API	Application Programming Interface
AR	Augmented Reality
CUDA	Compute Unified Device Architecture
FPS	Frames per second
GB	Gigabyte
GPU	Graphics Processing Unit
IoT	Internet of Things
MSAA	Multisample Anti-Aliasing
OLED	Organic Light Emitting Diode
OS	Operating System
PC	Personal Computer
PS	PlayStation
SDK	Software Development Kit
Unity	Cross-platform game engine
USB	Universal Serial Bus
VR	Virtual Reality
VRAM	Video Random Access Memory

1 INTRODUCTION

This thesis was commissioned by Nokia and for that reason the main requirements of this thesis work are from Nokia. The progress and steps of the thesis was planned together with Nokia.

Nokia is a global leader in developing network technologies. It is creating future technologies such as 5G-technology infrastructure for changing needs of human and the IoT. Nokia has employees with 160 different nationalities and operates in over 100 countries. (16.)

This thesis work is created in co-operation with OuluGarage. OuluGarage is a demo and innovation laboratory which provides room and tools for learning new things. Nokia offers co-operation to primary schools at OuluGarage where they can have multiple kinds of trainings. (15.)

This thesis work discusses mostly the Android mobile platform because of the market share of the Android operating system is biggest now as can be seen in the FIGURE 1 where Android holds 84,8% of market share in 2018. Android versions used in the thesis are Android 7.0 and Android 8.0.0.



Year	2016	2017	2018	2019	2020	2021	2022
Android	84,6%	85,1%	84,8%	85,2%	85,3%	85,4%	85,5%
iOS	14,7%	14,7%	15,1%	14,8%	14,6%	14,6%	14,5%
Others	0,7%	0,2%	0,1%	0,1%	0,1%	0,1%	0,1%
TOTAL	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

FIGURE 1. Mobile OS market share (10.)

I have most of my mobile application developing experience about the Android platform and Android development but Unity as a development tool was completely new to me. In Unity I used Microsoft Visual Studio as a coding development tool which I had used earlier for several different projects with several programming languages. Unity brings also several skill areas as its side-effect such as manipulating 3D models and designing 3D views which I had not done earlier.

This bachelors' thesis was commissioned by Nokia. The main target of the thesis required by Nokia was to study mobile platform possibilities, limitations, pros and cons in relation to the PC version of the VR application and to give an objective description of those in short and at least in a two-year time span.

The practical part of the work was to design and develop a mobile application of the VR PC application. The aim for the mobile app was to maintain such quality that it is suitable for customer communication such as expos and everyday marketing.

The devices used in this thesis work are Samsung Galaxy S8, Samsung Gear VR with a controller and Samsung Galaxy Tab A (2016). These devices were chosen because they were already present at Nokia.

2 WHAT IS VIRTUAL REALITY?

2.1 History of VR

Techniques based on Virtual Reality have existed for long time. Virtual Reality term was first used in 1938 when a French author described theater impression methods with words “la réalité virtuelle” which means Virtual Reality in English. In 1962 a machine called Sensorama used stereo sounds, fans and scents to enhance short film experiences. Sensorama which can be seen below in the FIGURE 2 provided Virtual Reality experience with techniques possible in its era. (1.)



FIGURE 2. Sensorama (1.)

Nowadays Virtual Reality becomes more and more common because different manufactures have started developing VR masks. Virtual reality is made possible by the progress of PC performance which has been improving all the time. In 2010 the biggest brands, such as Oculus, invested in the Virtual Reality development. In 2014 Facebook bought the Oculus corporation the with price tag of 1.7 billion euros. In that era Oculus was the leading developer of VR glasses. (1.)

2.2 VR in gaming

2.2.1 Game consoles

In 1995 Nintendo released Virtual Boy the first VR device for gaming. It was released in two markets; North America in August 14, 1995 and Japan July 21, 1995. The device itself had a dual screen design. Each screen had a monochromatic red LED display with a resolution of 382 x 224 pixel. These two displays allowed the user to see two different images at the same time which was how 3D illusion was made. Marketing of this device was not a huge success since there were only 770,000 units sold ever because this Nintendo discontinued Virtual Boy. (26.)

There are now many different manufacturers of Virtual Reality technologies and the technology is also used with game consoles. However, there are rather few companies using VR in the gaming segment. Only Sony has landed on non-PC VR game console markets with its PlayStation 4 game console. The PlayStation game console tracks the user movement with a camera and combines those with controller commands. This is how the user can affect game situations. (1.)

One of the biggest rivals of Sony PlayStations in the game console segment is Microsoft Xbox One, as can be seen in the FIGURE 3 below. Xbox One has been rumored to get Virtual Reality support somewhere soon. However, according to the article on the Verge, "Microsoft has completely pulled back on plans to support Virtual Reality headsets on the Xbox" which leaves PS 4 the only VR game console on the market. (23.)

PlayStation VR is compatible with the PS4 game console. Sony advertises PS VR with such features as 5.7-inch OLED display, 360-degree world and 120fps which should provide a smooth VR experience. PS VR has also 3D-sound and microphone for communication purposes. The user can control in game interactions with a Dualshock 4 controller, PlayStation Move or a PS VR aim controller. The PS VR system supports VR games, media and 3D Blu-ray. (27.)

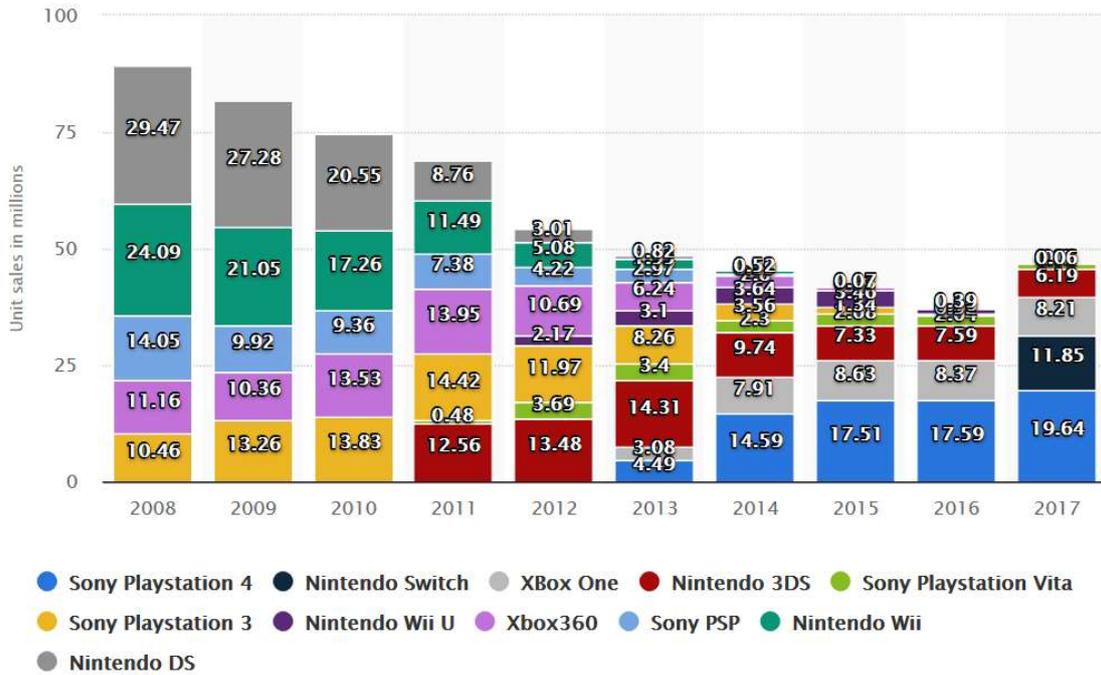


FIGURE 3. Global unit sales for game consoles from 2008 to 2017 (28.)

2.2.2 PC VR Gaming

Virtual Reality gaming can be a rather heavy operation for PCs and that is why there is also rather high hardware requirements. Nvidia recommends at least Nvidia GeForce GTX 1060 for desktops and laptops for the PC VR content (21.). The requirements are shown below in the FIGURE 4, which holds both desktop and laptop system requirements. The GeForce GTX 1060 3GB graphics card cost 199,99\$ at Nvidia's store on 15.1.2019 (37.).

SYSTEM REQUIREMENTS

What you need:

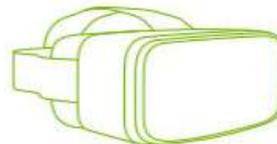


Desktop Requirements:

- **GPU:** NVIDIA GeForce GTX 1060 or greater
 - HTC Vive Pro GPU: NVIDIA GeForce GTX 1070 or greater
 - CPU: Intel Core i5- 4590 equivalent or greater
 - Memory/RAM: 8GB+ RAM
 - Video Output: 1x HDMI 1.3
 - Ports: 3x USB 3.0
- HTC Vive and Oculus Rift**
- OS: Windows 7 SP1 (64bit) or higher
 - Driver: Oculus – 361.91 and newer
HTC – 361.75 and newer
- Windows Mixed Reality**
- OS: Windows 10 Fall Creators Update
 - Driver: 387 and newer



NOTEBOOK
NVIDIA GeForce GTX 980



HMD
Head mounted Display

Notebook Requirements:

- **GPU:** GeForce GTX 1060 or greater*
 - HTC Vive Pro GPU: NVIDIA GeForce GTX 1070 or greater
 - CPU: Intel Core i5-7300HQ or greater
 - Memory/RAM: 8GB+ RAM
 - Video Output: 1x HDMI 1.3
 - Ports: 3x USB 3.0
- HTC Vive and Oculus Rift**
- OS: Windows 7 SP1 (64bit) or higher
 - Driver: Oculus – 361.91 and newer
HTC – 361.75 and newer
- Windows Mixed Reality**
- OS: Windows 10 Fall Creators Update
 - Driver: 387 and newer

*GeForce GTX 970M and 980M do not meet the minimum requirements.

FIGURE 4. Nvidia VR requirements (21.)

Virtual Reality games can be bought for PC from online game stores, e.g. Steam. Prices for the VR games vary from free to around 80€. (30.)

2.2.3 Mobile VR Gaming

When Google launched a Virtual Reality headset for smartphones for price starting at 5.71 USD, consumers had access to the VR content cheap. Google also provides the VR content for Android devices at the Google Play, store which is Google's app store for Android. (8.)

There are multiple free and paid games available for Android users in different gaming genres. The user can either choose amongst other things between FPS, racing, adventure and puzzle games.

Most headsets support the screen size from 4 inches up to 6 inches but some might support even 7-inch screen smartphones. A better resolution will enable smoother visual experience and prevent a clearly visible pixelation. For example, the Samsung flagship phone Galaxy S9 has 2960 x 1440 pixel resolution. (12.)

There is also software from Riftcat called VRidge which enables the user to turn a phone into a PC VR headset. Currently the software supports Android phones with Android 5.0 or newer OS, and it is also possible to use standalone headsets. Supported headsets are HTC VIVE Focus, Oculus Go, Samsung Gear VR and Google Daydream View. This enables users to experience the VR content with a PC performance at low price. (22.)

3 MOBILE VR

Mobile VR is described most accurately in the thesis work because of its focus on studying mobile platform possibilities in the VR context. Android is chosen as main study target because of its popularity in the field of mobile platforms, which can be seen in the FIGURE 1.

3.1 History

Google has launched its Google Cardboard VR headset shown in the FIGURE 5 in UK in June 2014. This led to the point where Google focused more on the VR content, such as 360-degree videos, VR apps and games. Cardboard had a huge market share in 2016 as can be seen in the FIGURE 6 where the Cardboard takes 69% and the second comes Samsung Gear VR with a 17% share. Cardboard design is very simple. It is made from cardboard which houses a couple of lenses and magnets acting as hardware buttons. (2.)



FIGURE 5. Google Cardboard (7.)

Samsung is known for its high-end mobile phones and it has developed its own VR glasses for phones with Oculus. For mobile VR headsets the mobile phone is acting as a display. Samsung VR glasses are called Samsung Gear. (1.)

Samsung Gear VR is compatible with Samsung manufactured flagship mobile phones. It weighs 345g, has a 101-degree field of view and a changeable USB interface which supports Micro USB and USB Type-C. Its advantage is that Samsung has developed Samsung Gear VR with Oculus and they came up with a dedicated controller for Samsung Gear VR. The controller provides many interaction possibilities with its G-sensor, touchpad and multiple buttons. (24.)

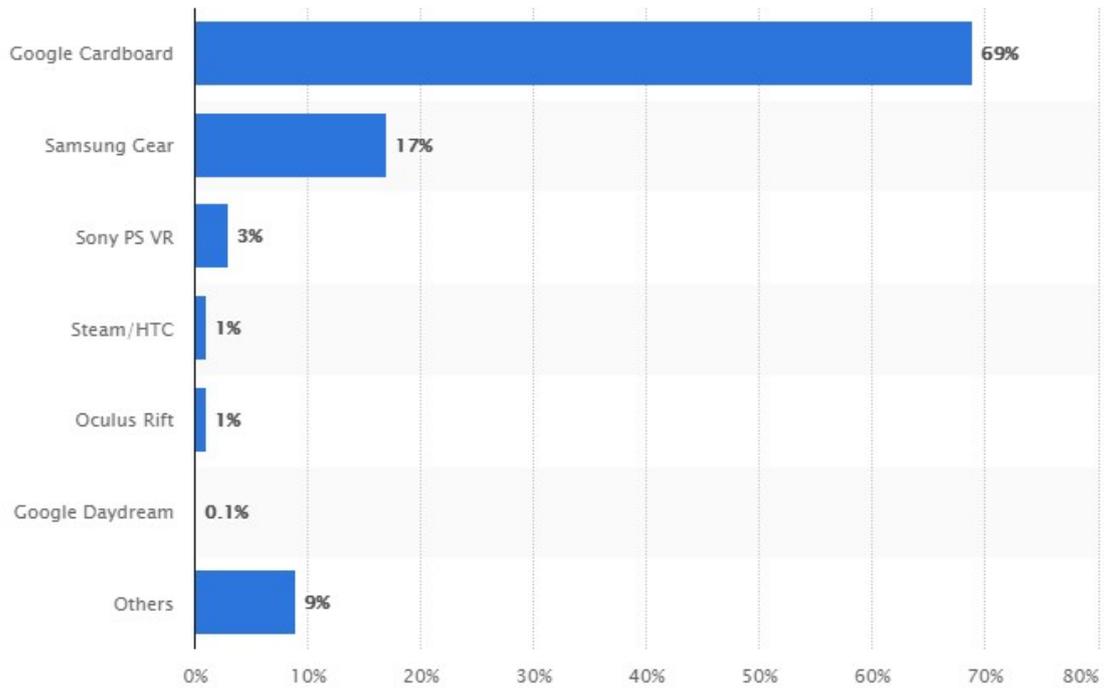


FIGURE 6. Global VR headset unit shipments market share by vendor in 2016 (29.)

3.2 System requirements

Minimum and recommended hardware and software requirements for Google VR SDK are described here. This section describes only mobile device requirements not requirements to develop VR applications. The user should at least have the Android API SDK version 21 installed. It is also known as Android Lollipop or Android L. Google VR application developing requires a Cardboard or Daydream supported phone. Unity has listed minimum and recommended hardware and software requirements which can be seen in the FIGURE 7 below. Samsung Gear VR requirements can be seen in the FIGURE 8 in Combability section.

Minimum hardware and software requirements

This section provides an overview of the minimum hardware and software requirements for developing with the Google VR SDK.

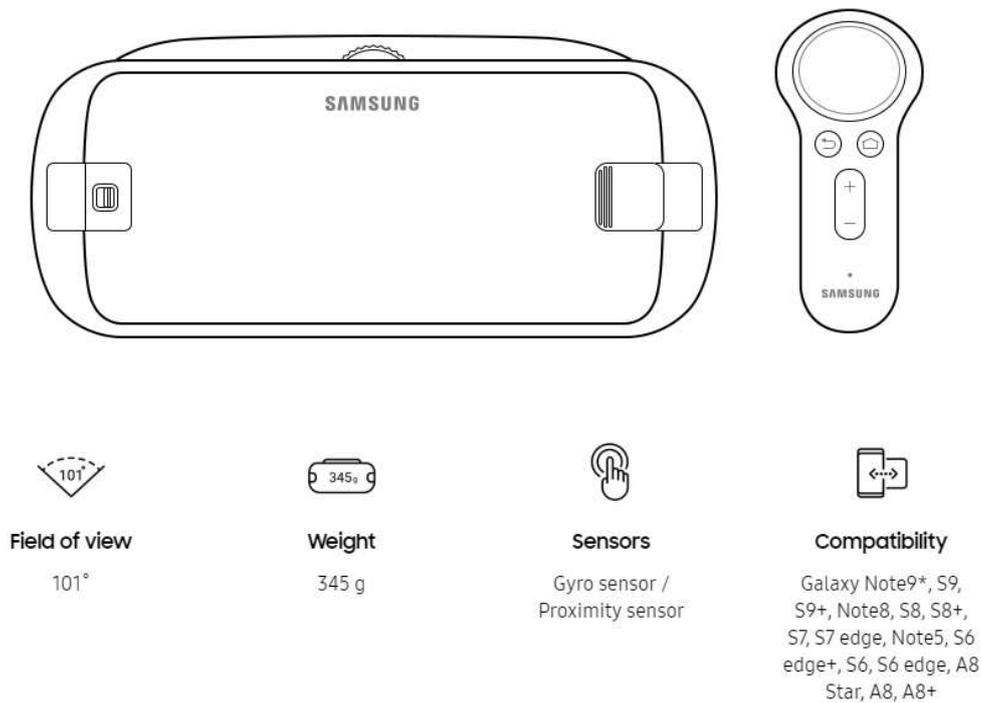
Device	Hardware	Software
Cardboard	<ul style="list-style-type: none"> - Devices running Android 4.1 or later with a gyroscope. - Devices running iOS 8 or later with a gyroscope. - A viewing device compatible with Cardboard. 	<ul style="list-style-type: none"> - Unity Cardboard integration requires Android Lollipop or greater. - iOS devices with the Google Cardboard application installed.
Daydream	<ul style="list-style-type: none"> - Daydream compatible devices. See the Google Daydream Hardware page for a list of devices compatible with Daydream. 	

Recommended hardware and software requirements

This section provides an overview of the recommended hardware and software requirements for developing with the Google VR SDK.

Device	Hardware	Software
Cardboard	<ul style="list-style-type: none"> - For Android, refer to the documentation provided with your Cardboard viewer for a full list of compatible devices. - For iOS, any iPhone 5 series or later phone. - A viewing device compatible with Cardboard. 	<ul style="list-style-type: none"> - Android 5.0 or later. - iOS 8 or later, with the Google Cardboard application installed.
Daydream	<ul style="list-style-type: none"> - Any device compatible with the Daydream specification. See the Google Daydream Hardware page for a list of devices compatible with Daydream. 	<ul style="list-style-type: none"> - Android 7 or later.

FIGURE 7. Mobile VR requirements for Cardboard and Daydream (31.)



*Galaxy Note9 is compatible with Gear VR model number SM-R325NZVC*** only.

FIGURE 8. Samsung Gear VR compatibility (24.)

Developing with Unity to Samsung Gear VR requires Oculus SDK provided by Oculus which can be integrated into Unity. SDK includes for example controller support and button mappings for different controllers, such as the Xbox One controller and Gear VR controller.

3.3 Controllers

This section is telling what kinds of controlling method can be used to achieve wanted actions in the mobile Virtual Reality.

3.3.1 Supported controllers

Oculus SDK supports a Samsung Gear VR controller and XBOX One controller officially and has instructions how to use a supported controller in a Unity project. Controller buttons can simply be mapped via an OVRInput class which is included in Oculus Unity SDK. (6.)

The Android mobile operating system itself supports a lot of different controllers using USB or Bluetooth interfaces. Android has been supporting game controllers since Android API 9. (4.)

Connecting PS 4 controller to Android mobile device was tested. It can be done quite fast from Android settings. The controller connects to a phone via the Bluetooth interface. After all a PS 4 controller had noticeable input lag in relation to a Samsung Gear VR controller which worked flawlessly. The input lag problem is not only related to the devices used because there are multiple blog posts describing solutions to fix the problem. For reference a web page called maketecheasier recommends users to download an app which can help reducing the input lag (35.).

An Xbox One controller was also tested which should have been compatible with Android, but it came up that the controller was not a Bluetooth controller and for that reason not connectable to an Android device wirelessly. The Bluetooth controller can be defined whether it has a certain kind of Xbox button or not. If the plastic frame of the Xbox button is part of the face of the controller, the controller supports Bluetooth and otherwise not. (14.)

Samsung has developed its own controller for VR purposes with Oculus. The controller is small, and it looks a lot like a TV remote controller as can be seen in the FIGURE 9 below. It has quite

few buttons. There is a touchpad for swipes, a back key and a home key, a volume key for controlling volume and a trigger key. Some of the keys are not programmable when developing a Unity project. These keys are marked as reserved in the FIGURE 9. The controller has sensors that track rotation and it rotates the controller in the VR but there is no position tracking so the user cannot reach out with it in the Virtual Reality. Gear VR controller is powered with two AAA batteries and they are located the bottom of the controller. The controller connects to the device via Bluetooth. (25.)

Gear VR Controller – Virtual Mappings

(Typical Application)

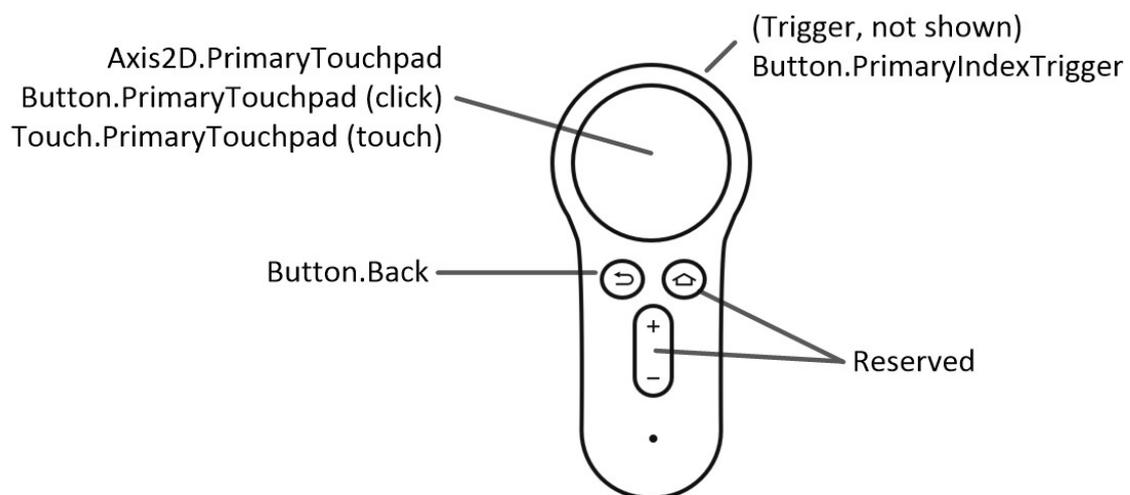


FIGURE 9. Samsung Gear VR controller button mapping (6.)

3.3.2 Other possible controlling methods

Gear VR does not necessarily need an external controller because it can be controlled with voice commands. The voice controlling is called Oculus Voice which is available for Gear VR users. Oculus Voice enables the user to launch actions with specific voice commands. (11.)

The user must enable Oculus Voice for once and then it is usable thereafter. With Oculus Voice user can perform certain actions, but it is not capable of doing every action. Oculus Voice starts when the user says “Okay Oculus”. After that there is a screen which shows voice commands that can be triggered. (11.)

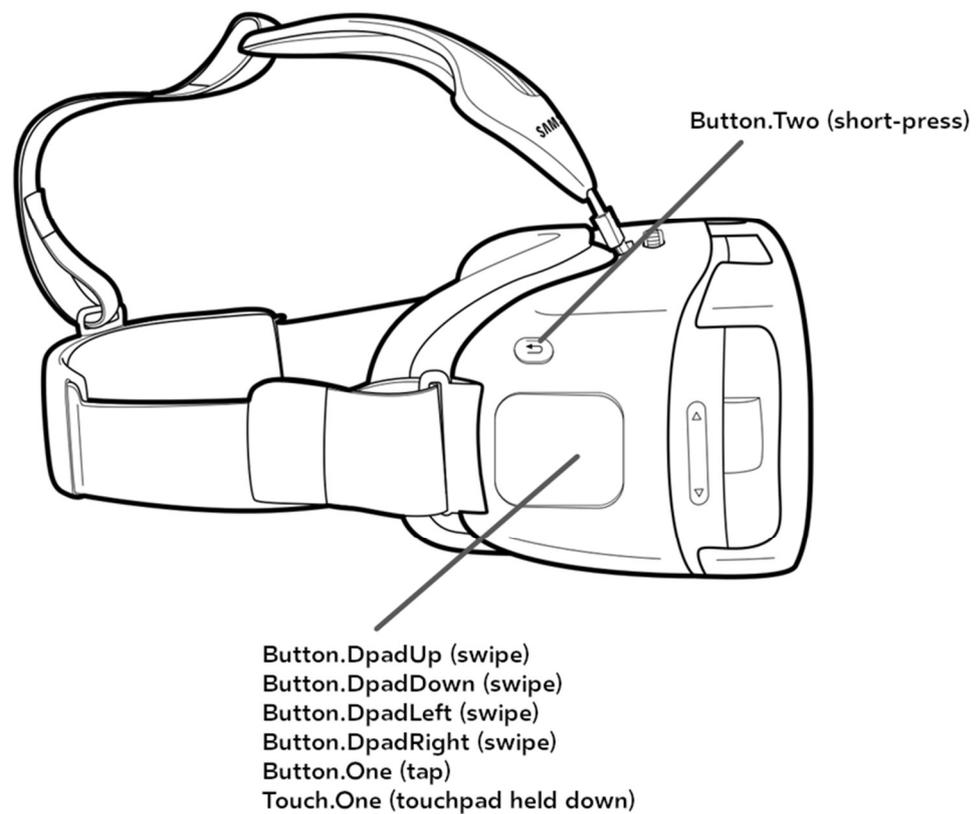


FIGURE 10. Samsung Gear VR headset (6.)

Samsung Gear VR headset has a similar touchpad than laptops have as a mouse. The touchpad is located the right side of the headset. The touchpad can be mapped with Unity to do some custom actions in the VR content, such as moving or menu actions. The touchpad location is demonstrated in the FIGURE 10 and there is also a button called “Button.Two” which can be mapped to custom action. Including basic buttons on the headset, the headset enables to experience the VR content without having any other external accessories in addition to a headset. (6.)

3.4 Multiplayer in mobile VR



FIGURE 11. Samsung Gear VR multiplayer game (24.)

At the start of this chapter there had to be questions described that needed to be studied. The questions consider only Android OS. Those questions are described as chapters below:

3.4.1 Existing application examples

There is a lot of VR content also for mobile platforms and these can be downloaded from app stores such as Google Play. There is for example Cardboard Camera which is basically a camera application that can be used with the Google Cardboard headset, Google Street View which can be used to see 360-degree views of different places around the world and Cardboard Theater which is video player for Google Cardboard. There is rather wide variety of VR applications available. (9.)

3.4.2 Multiplayer player amount

There is a game called Social Trivia (by Oculus) which can host up to five players (13.). Samsung Gear VR has already some multiplayer VR games that users can buy from Oculus' store. Some of them are designed for two players and some are up to four players. Because there are already

multiplayer games available, it can be deduced that it is possible to create multiplayer features in mobile VR applications. (5.)

3.4.3 Hardware requirements

The user needs a smartphone which should be modern and has a tolerable resolution. The smartphone should have a G-sensor and it should support either Google Cardboard or Google Daydream. Also, the Google Cardboard or Google Daydream supported headset is required. (12.)

There are more requirements listed at the chapter 3.2.

3.4.4 Software requirements

The user needs a device with support of either the Google Cardboard or Google Daydream (12.). More specific software requirements can be found at the chapter 3.2.

3.5 Hardware performance comparison between PC and mobile devices

3.5.1 GPU comparison

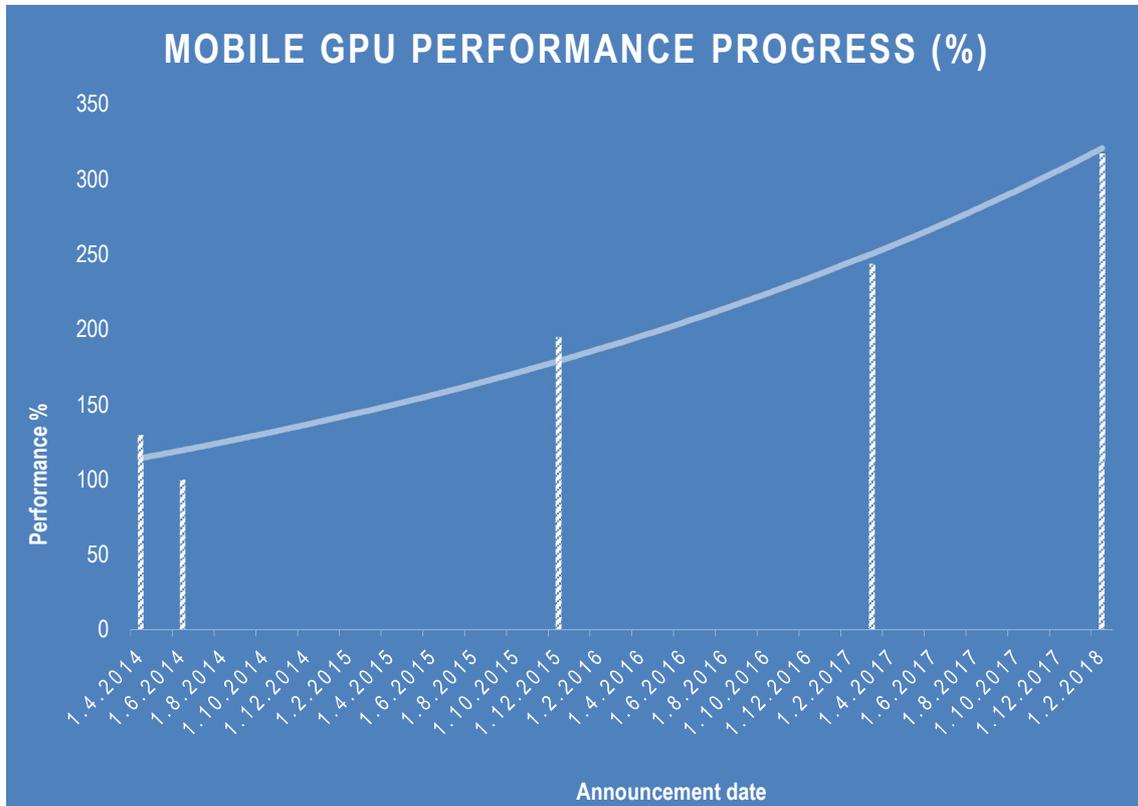


FIGURE 12. Mobile Adreno GPU progress on performance

The FIGURE 7 is produced with Qualcomm's own statements of their mobile GPU performance. They described the performance of new series by comparing it percentually to their older series and the FIGURE 7 is based on those statements.

Qualcomm Adreno 540 is one of the high-end GPUs present nowadays. The GPU was introduced in early 2017 and it has an integrated graphics card of Snapdragon 835 SoC which, is produced for smartphones. The SoC can be found in top-notch smartphones running the Android operating system, such as OnePlus 5T and Nokia 8 Sirocco. (18.)

Nvidia GeForce GTX 1060 is one of the lowest GPUs of Nvidia 10-series for gaming purposes. There are 3GB VRAM and 6GB VRAM versions of the card with the speed of 8Gbps. The 6GB version has 1280 and 3GB version has 1152 CUDA Cores. Nvidia claims that with GTX 1060 it is possible to use the VR content. (19.)

Notebookcheck has done a comparison chart to visualize differences between these two graphic cards even though they are used with different platforms. The PC platform has many benefits that the mobile one has not, such as better cooling solutions, bigger power supplies and a bigger physical size. These benefits can be seen taking an advantage over the mobile graphic card in the FIGURE 13.

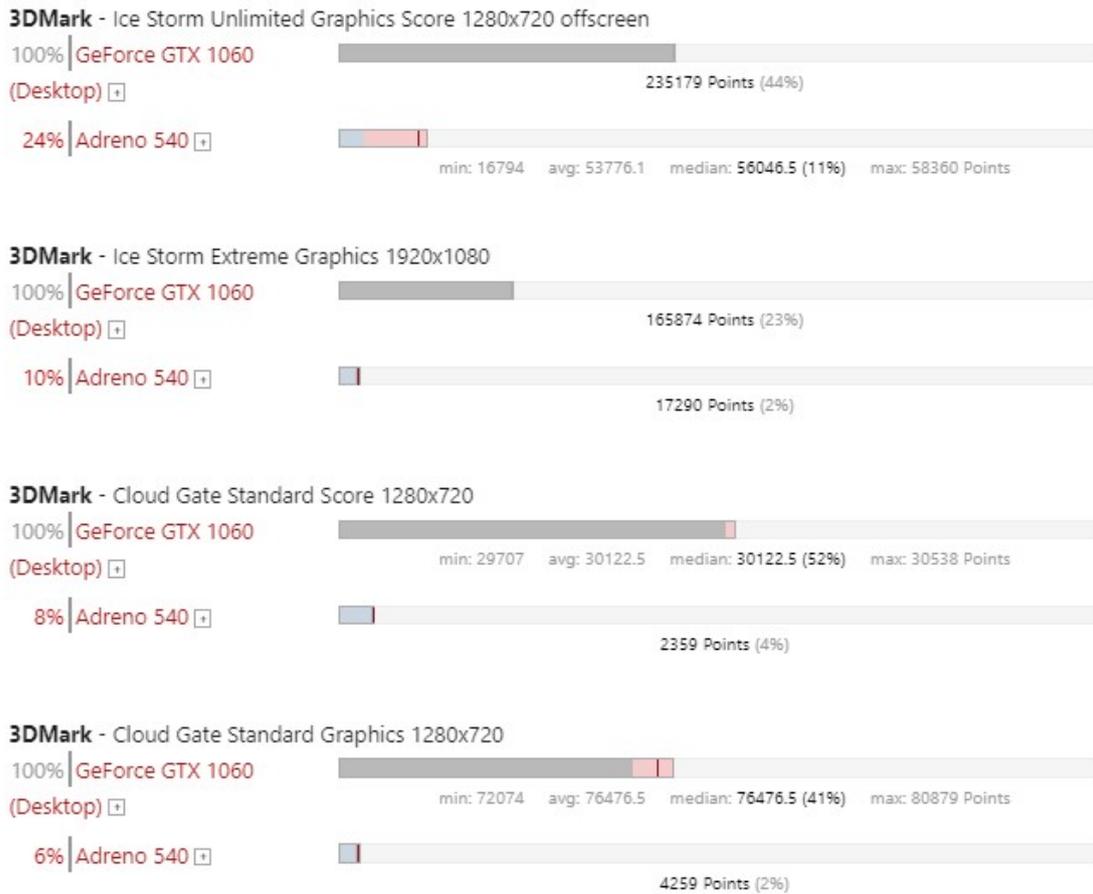


FIGURE 13. Comparison between Adreno 540 and GTX 1060 GPUs (18.)

3.6 Porting Unity PC application to mobile platforms

There are a lot of multiplatform development tools, such as Unity which was chosen because it is already in use at Nokia. Unity enables to develop for over 25 platforms, such as Android, iOS, Gear VR. (34.)

3.6.1 Porting to Android platform

Unity supports many platforms including the most mainstream mobile platform Android. Porting to readymade PC Unity application to Android however has some requirements.

Unity projects developed to the PC platform can be rather heavy for mobile devices and therefore they should be optimized for the target platform. Scenes built with Unity can be rather heavy for mobile devices to consume because of complex 3D models on the scene. To achieve a tolerable framerate, it is recommendable to create less complex 3D models for mobile use and replace the complex ones. The 3D models can be lightened with Blender which is a PC tool to handle models. One way to lighten the models is to use Blender's decimate tool which reduces the model complexity by reducing its vertices.

Optimizations can help to maintain a tolerable framerate and, in that way, to ensure a smooth VR experience. To optimize the PC application for a mobile use furthermore, it is also recommended to pay attention to lightning and shadows which need lots of performance. The smoother VR experience can be achieved e.g. by disabling real-time shadows, creating light 3D models, disabling MSAA and by setting a texture resolution to eight-of-res.

There is no quick way to do all the optimizations, but Unity has created a simple checklist to maintain the framerate in games. Following the checklist in the FIGURE 14 during the whole developing process helps to maintain a smooth experience.

Simple checklist to make your game faster

- Keep the vertex count below 200K and 3M per frame when building for PC (depending on the target GPU).
- If you're using built-in shaders, pick ones from the **Mobile** or **Unlit** categories. They work on non-mobile platforms as well, but are simplified and approximated versions of the more complex shaders.
- Keep the number of different materials per scene low, and share as many materials between different objects as possible.
- Set the **Static** property on a non-moving object to allow internal optimizations like [static batching](#).
- Only have a single (preferably directional) **pixel light** affecting your geometry, rather than multiples.
- Bake lighting rather than using dynamic lighting.
- Use compressed texture formats when possible, and use 16-bit textures over 32-bit textures.
- Avoid using fog where possible.
- Use [Occlusion Culling](#) to reduce the amount of visible geometry and draw-calls in cases of complex static scenes with lots of occlusion. Design your levels with occlusion culling in mind.
- Use skyboxes to "fake" distant geometry.
- Use pixel shaders or texture combiners to mix several textures instead of a multi-pass approach.
- Use **half** precision variables where possible.
- Minimize use of complex mathematical operations such as **pow**, **sin** and **cos** in pixel shaders.
- Use fewer textures per fragment.

FIGURE 14. Unity's checklist to make game faster (33.)

3.6.2 Porting to iOS platform

Unity also supports the second popular mobile operating system iOS, which is known as the operating system of Apple mobile phones and tablets.

Developing for iOS has some requirements which need to be met. The operating system must be Macintosh as it is Apple's operating system for desktop and laptop devices. Macintosh is needed because of XCode, which is a development tool for iOS and can be used to build projects for iOS. The Unity version at least 5.3.4 or later is needed to create the Unity project and an iOS device to be the target device to build for. (32.)

When Unity is used to build applications to the iOS platform the user has to go through several crucial steps. The first step is to install Unity which is at least version 5.3.4. The very next step is to install the XCode which is a developing tool for the iOS. Then the user has to create a new project that is going to be developed and switch a deployment platform to the iOS in the Unity's build settings. Next there is the developing part where the user has to create functionality for the application. After these steps the project can be built and opened to the XCode. The user has to add Apple ID to XCode and sign application with team of Apple ID before the project can be deployed with the XCode. Next a deployable device has to be selected from the XCode's settings. The final step is to press a build and run button from XCode. (32.)

Tweaking a Unity project to the iOS platform is very much similar to that of Android. Optimization methods can be almost directly taken from Android tweak methods as described in the chapter 3.6.1. It is very important to pay attention to real-time shadows, lightning and the quality of 3D models.

4 PRACTICAL PART

The thesis work started by starting to study Unity features and how to create simple Android application with Unity. First, there had to be studied requirements to build android application with Unity. Then the thesis work had to continue in short iterations where the author first wanted to implement some feature and then had to study from several sources how the feature could be implemented. The short iteration style of studying continued through the whole study.

4.1 Mobile Virtual Reality application

Because of the lack of Unity experience the work had to be started from the beginning. First feature was to manipulate a 3D model position with a Samsung Gear VR controller. Next feature added dimensions to controlling the 3D model on the screen. First dimensions were defined by camera pointing direction which was rotated by the head movement, but later dimensions were taken from a controller rotation and user's finger position on a touchpad of the controller.

After reaching the basic knowledge of how to manipulate the 3D object position and rotation on the scene, the work continued by simple performance tests. These tests were implemented by creating a scene where the user can add complex 3D objects on the scene by pressing a button on the controller. The effects of adding complex 3D objects on the scene were tracked by monitoring a fps counter on the screen.

After that the next milestone was to start creating VR application for the Android mobile platform which had quite similar controlling methods as the VR application for the PC platform. Starting to port the PC VR application was easy to start with similar controlling methods and similar handling of a surrounding world where the user is in a centrum of everything. In this kind of mobile VR app the user can interact with the surrounding world by using a controller dedicated for Samsung Gear VR. Flying, grabbing objects and teleporting could be done with one controller which was implemented practically by creating tools to Virtual Reality.

Each tool had one kind of interaction with the surrounding world. A jetpack tool could be used for flying faster than with a default tool. A hand tool could be used to grab interactable objects and a

default tool had all basic functions such as flying and teleporting. All the tools had different models. The jetpack tool looked like a hold able rocket, the hand tool looked like a hand and the default tool was shaped like a controller the user had in the real world. Practically every tool had a shape telling the user what kinds of interactions it is able to do. The shapes of the tools can be seen in the FIGURE 15 which is a screenshot of the Mobile Gear VR application that was created during the study.

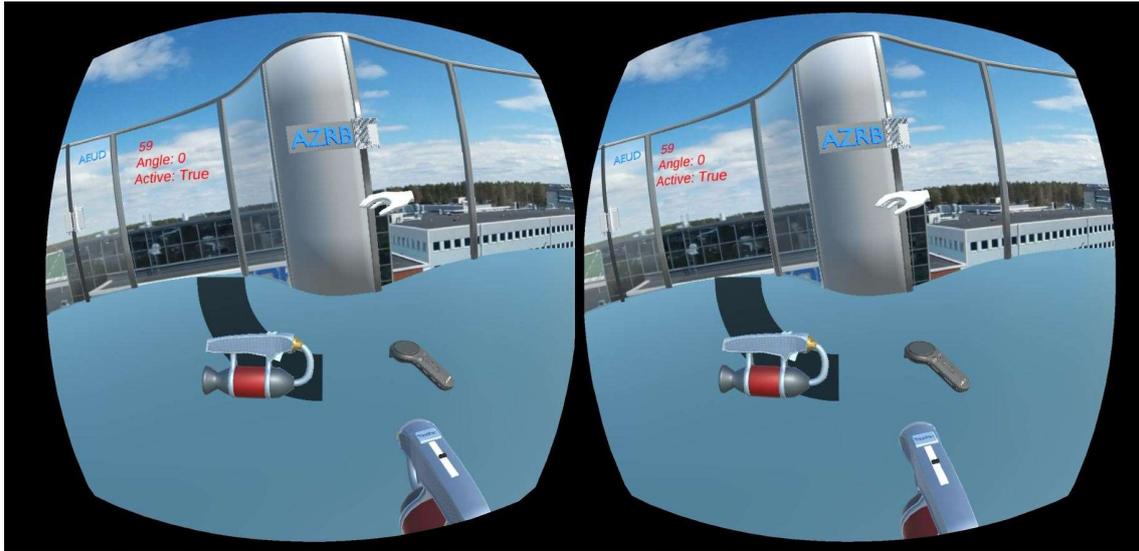


FIGURE 15. Mobile VR application toolbox

The teleporting tool looked like the controller the user was holding in the real world also. The only difference was that the tool casted beam when activated. When the beam had hit a point on an allowed teleport platform the user could just press a teleport button to get in to that place. The teleport point was visualized with a green holographic sphere as seen in the FIGURE 16.

The mobile VR application tried to be a mobile version of the already created PC version. The PC version had all these features: grabbing items, flying and teleporting. It felt logical to start developing the mobile version similar than the PC version with similar controlling methods and the VR platform. After all, mobile devices had very limited performance, the prices of top-notch devices were quite high, controlling methods were at the same level with PC hardware and overall user experience was worse. The only advantage the mobile platform has over the PC platform is its physical size. Mobile devices are quite small compared to PC devices.

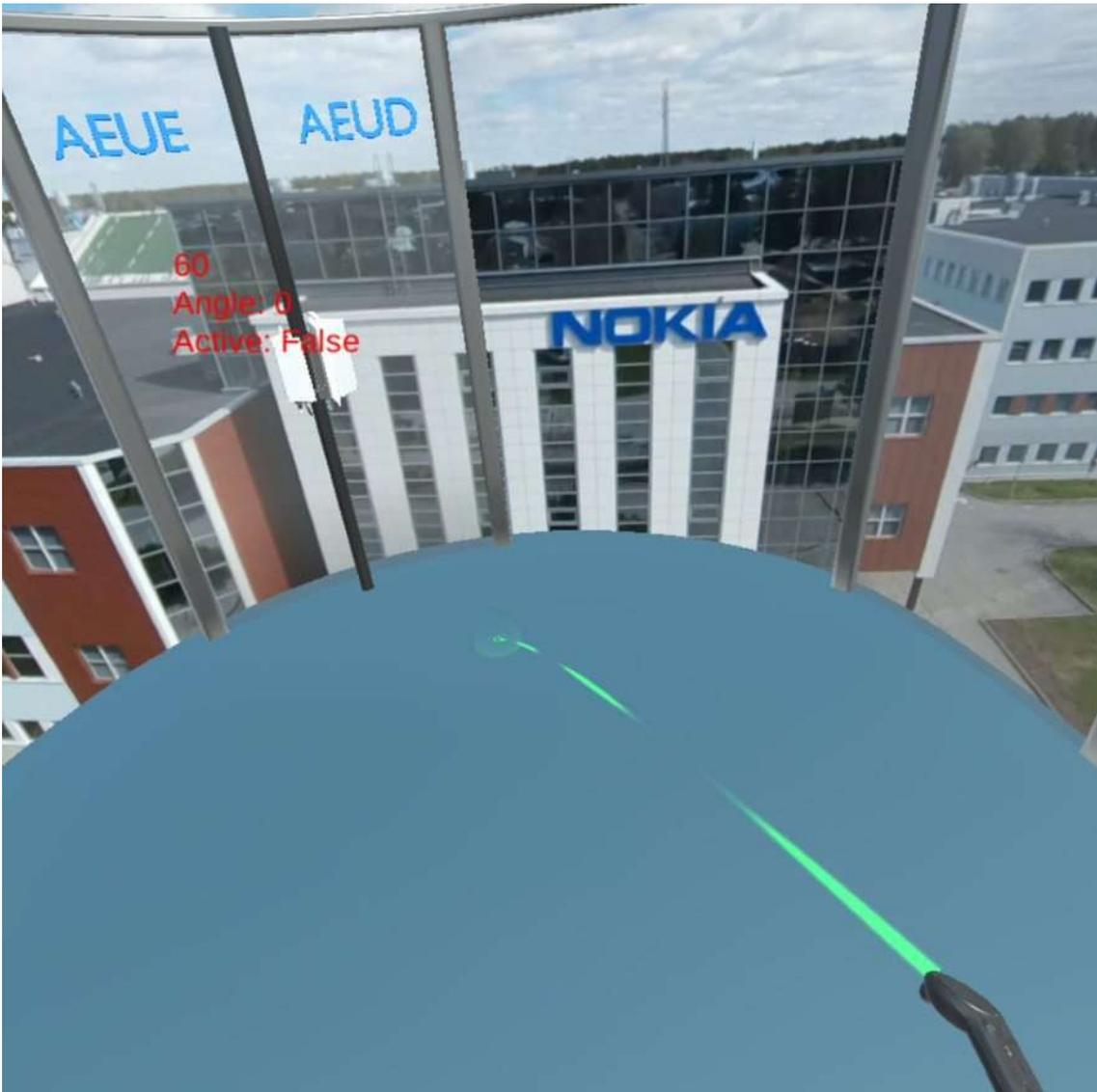


FIGURE 16. Teleport tool in mobile VR application

The requirements for the mobile application were that it could be used for customer communication in situations like expos and everyday marketing. These requirements are filled better with a flat screen typical mobile application than the mobile VR application. The mobile VR application's weakness is that every viewer should have their own mobile device and a VR mask to use or view the VR content. Because the VR the user is isolated from the real world. It is quite hard to demonstrate anything to the user in a controllable way. With the typical flat screen application, a demonstrator and a customer can share a same screen and the demonstrator can manipulate the application view at the same time.

With the current PC hardware the user can crouch, which will affect the Virtual Reality world but this feature is not possible yet with the mobile VR platform or could not be easily implemented. The

PC version had support for dual controllers which allowed many new interactions and it also allowed the user to use two different tools at the same time. The user can also reach out items in VR, but this is not supported with the Samsung Gear VR system. The lack of crucial features led to abandon the mobile VR platform with this project.

After all a usable VR showroom for the company's products was created during the work, which had multiple products, multiple VR tools and a framerate of 60 frames-per-second. The FIGURE 17 below is the editor view of the application. On left hand the user has a hierarchy view of objects on the scene showed visually in the middle of the editor.

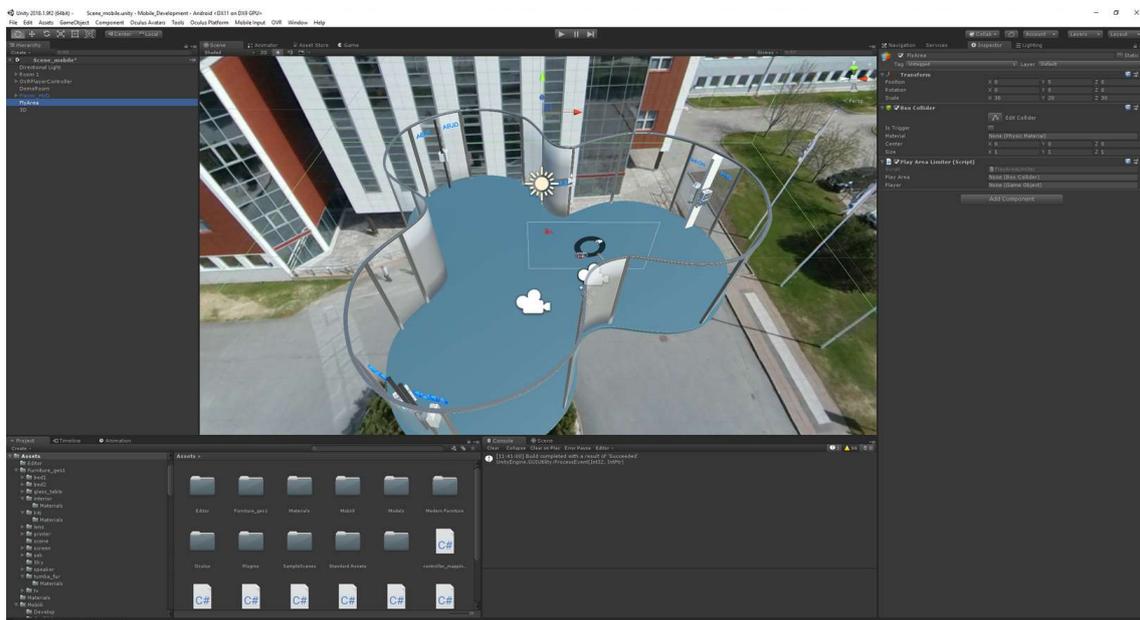


FIGURE 17. Mobile VR application in Unity editor

4.2 Flat screen mobile application

After abandoning the idea of porting the VR application directly from the PC platform to the mobile platform it was decided to lead developing to the flat screen 3D application. This meant a different kind of handling of the Virtual world. In the VR world surrounds the user and the user can affect the view by turning head but the flat screen application needs to be implemented differently. The application had to bring on different ways to affect to the world and different kinds of controlling methods. Finally, it led to developing a touch input on the app where the user can manipulate a camera viewing angle and position by different touch gestures.

Controlling methods of the flat screen applications differ from controlling methods of the VR application in a way that there is no external controller, but controlling is implemented purely by a touchscreen and gestures. Moving is implemented by a pinch gesture where the user must use two fingers to zoom in or out from the virtual world. The zoom moving point is the center point between the user's fingers and in that way, the user can manipulate distance and position at the same time. By swiping on the touchscreen the user can manipulate the position in two dimensions: x- and z-axes. It occurs to the user as moving forwards and sideways.

The flat screen application is easier to use compared to the PC VR version in some situations. For example, when the user wants to pre-plan a network coverage with the PC VR version, antennas need to be first reached and then dragged to a wanted point. The flat screen application does this much easier with just two taps on the screen. First the user selects a wanted antenna type and then just taps on a point where it should be attached. For this kinds of use cases the touchscreen serves user's needs better than the VR environment could do.

First created flat screen application had a showroom which had products of Nokia and their specs showing on the walls next to each product. Moving was possible with the pinch zoom gesture and it was possible to view products closer by tapping on them. The application showed a product in front of the user when tapped on the product. It was also able to rotate the product in any direction when in review mode. The application was created to only demonstrate how the idea from the PC VR application could be implemented in the mobile flat screen application.

Next, application that could be used to pre-plan network coverage was created. The application had a product catalog which contains Nokia antenna products where the user could choose an antenna to be attached in city view. After choosing the antenna, the application shows attachment points where the user can attach the current antenna. Attachment points are showing as yellow spheres on the scene as seen in the FIGURE 18 and there can be also seen a blue colored beam coming from a mast on the right. Then the user can rotate an antenna beam, which is visualized through a cone shaped almost transparent beam object. Similar features were already implemented in the PC VR version where the user had to drag antennas to the positions. The mobile version of the network coverage pre-planning is for demo purposes only and it does not calculate the real network coverage on the area.

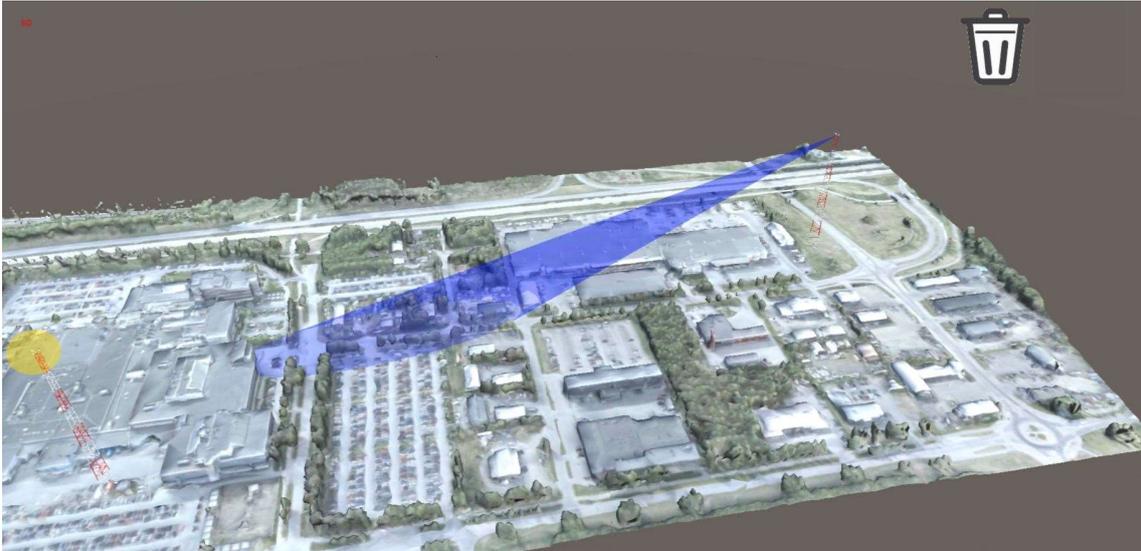


FIGURE 18. Mobile flat screen network coverage design application

Attaching an object to the mast was implemented by using a raycast. Ray was casted straight from the point the user touched on the screen. Then if the ray hits an object the application had to check if an object has a component called Attachable and if it did, the application had to call OnAttach function via SendMessage function. The OnAttach function includes all functionality to manage actual attaching. A code part for the ray casting operation can be seen in an attachment AttachObject.

4.3 Mobile and pc platform pros and cons in Virtual Reality

The Table 1 below visualizes the features of different platforms. The table 1 is depending on current hardware used in this thesis work. Comparing is done between Samsung Galaxy S8, Samsung Gear VR and HTC VIVE.

TABLE 1 Platform comparison

Feature	PC VR	Mobile VR	Mobile Flat Screen
Multiple spectators, same device ¹	✗	✗	✓
Controller support	✓	✓	✓
Controller required	✓	✓ ²	✗
Controller position tracking	✓	✗	✗
Controller rotation tracking	✓	✓	✗
VR headset required	✓	✓	✗
Wireless	✗	✓	✓
Multiplayer	✓	✓	✓
Isolated from real world ³	✓	✓	✗
3D audio	✓	✗	✗
Dual controller support	✓	✗	✗

¹ Same device means visible for multiple users with one display/headset

² Samsung Gear VR headset has integrated touchpad and buttons

³ User has blocked field of vision

5 RESULTS

This chapter describes a comparison made between the already made PC VR application and the mobile applications I made during the thesis work.

First of all, the mobile VR application caused motion sickness for several people because the lack of performance caused a bad framerate. Motion sickness however depends on the user since it does not occur for everyone. The bad framerate may be also caused by the case that the application was not optimized enough for the hardware. The motion sickness problem was not present as strongly with the PC VR application as with the mobile VR application.

Mobile devices had much worse resolution which made the mobile VR application look much worse than the PC VR application used with HTC VIVE or HTC VIVE Pro. This was more a hardware problem than based on software.

As the Samsung Gear VR controller lacks the feature to reach which means that it does not track the controller position, the mobile VR version also lacks similar object grab feature as the PC VR version did. However, the mobile VR version had an object grab version but the user was not able to bring the object closer or remove it farther by moving the controller. The user was only able to rotate the object by rotating the controller.

After all, there were three different mobile applications. Two of them were regular flat screen applications that had a touch screen user interface. The network coverage pre-planning application was more intuitive on the mobile touch user interface than in PC VR where the user had to first reach an antenna and then drag and drop it on its installation place. The same feature was implemented with touch screen user interface by only tapping on the antenna in demand and then tapping on the installation position indicated by a glowing sphere as can be seen in the FIGURE 18. In the PC VR version the user had to reach a beam of antenna by reaching it and then dragging it by the controller. This feature was also more logical with touch screen user interface where the user could just drag it by tapping on it and then dragging it the position. However, these statements are based on the author's subjective experience of both applications and cannot be interpreted objective facts.

6 CONCLUSION

The main target was to study mobile VR possibilities as a platform so that the company that commissioned the thesis work could get an objective point of view about the possibilities for their own needs.

The company did not find the mobile VR platform good enough to be used in their needs in the current state of the platform which can be considered as the main result of the thesis work. Instead of seeing potential in mobile VR, they saw more potential in an ordinary touch screen mobile application. The company's point of view was based on my studies and on the applications made during the study.

During this thesis work, I studied the topic iteratively by studying first and then implementing the studied subject in practice in the mobile applications I made. This way of studying and realizing in practice continued during the whole work and I had also demos to show in practice for the supervisor from the company.

I consider my work to be a success from the point of view that I expanded my very own and the company's knowledge of the possibilities, limitations, pros and cons of the mobile VR platform. I used tools which were not familiar to me and because of that, I learned a lot of the topic and the tools I used. The company that commissioned this thesis work was gratified with the result.

This thesis work could be extended by expanding studied mobile platforms to iOS and by studying AR besides of VR. Studying iOS could be useful because it is the most popular OS right after Android in the FIGURE 1. Also, because of the lack of performance in mobile devices, it is a good idea to study how to do heavy calculations on the server side and in that way reduce a load on a mobile device processor.

REFERENCES

1. Arvanaghi, Babak – Skytt, Lasse 2016. Virtuaalitodellisuus – tulevaisuus on täällä tänään. Date of retrieval 4.11.2018. <https://tieku.fi/teknologia/vempaimet/virtuaalitodellisuus>
2. Bell, Lee 2016. Google Cardboard launches in UK for £15. Date of retrieval 10.11.2018 <https://www.wired.co.uk/article/google-cardboard-vr-headset-uk-launch>
3. Developers 2018. Support game controllers. Date of retrieval 22.11.2018. <https://developer.android.com/training/game-controllers/>
4. Developers. Support game controllers. Date of retrieval 19.1.2019. <https://developer.android.com/training/game-controllers/>
5. Endicott, Sean 2017. Best multiplayer games for Gear VR. Date of retrieval 21.1.2019. <https://www.vrheads.com/best-multiplayer-games-gear-vr>
6. Facebook Technologies, LLC. OVRInput. Date of retrieval 29.11.2018. <https://developer.oculus.com/documentation/unity/latest/concepts/unity-ovrinput/>
7. Google Corporation. Google Cardboard. Date of retrieval 15.12.2018. <https://vr.google.com/cardboard/>
8. Google Corporation. Oma Cardboard. Date of retrieval 9.1.2019. <https://vr.google.com/intl/fi-fi/cardboard/get-cardboard/>
9. Hindy, Joe 2018. 10 best Google Cardboard apps for the best VR experience!. Date of retrieval 21.1.2019. <https://www.androidauthority.com/best-google-cardboard-vr-apps-622766/>
10. IDC 2018. Smartphone OS Market Share. Date of retrieval 26.11.2018. <https://www.idc.com/promo/smartphone-market-share/os>
11. Karner, Jen 2017. How to use Oculus Voice. Date of retrieval 28.11.2018. https://www.vrheads.com/how-use-oculus-voice?_ga=2.75264146.1690986791.1543402931-1057630105.1525698085
12. Kuryliak, Oleksiy 2018. Everything You Need to Enjoy VR On Mobile in 2018. Date of retrieval 21.1.2019. <https://virtualrealitypop.com/everything-you-need-to-enjoy-vr-on-mobile-in-2018-dca39a004385>
13. Lang, Ben 2016. Oculus Bringing New Multiplayer Games and Social Features to Gear VR. Date of retrieval 21.1.2019. <https://www.roadtovr.com/new-samsung-gear-vr-social-features-oculus-social-trivia-herobound-gladiators-multiplayer-facebook-360-video/>

14. Microsoft. Use your Xbox Wireless Controller on Samsung Gear VR. Date of retrieval 15.1.2019. <https://support.xbox.com/en-US/xbox-one/accessories/use-samsung-gear-vr-with-xbox-controller>
15. Nokia Corporation. Creating the future. Date of retrieval 3.11.2018 <https://www.nokia.com/fi-fi/node/58556>
16. Nokia Corporation. Ohjelmoitavan maailman globaali teknologiajohtaja. Date of retrieval 3.11.2018 <https://www.nokia.com/fi-fi/tietoa-meista/keita-olemme>
17. Nokia Corporation. Our history. Date of retrieval 3.11.2018 <https://www.nokia.com/en-int/about-nokia/who-we-are/our-history>
18. NoteBookcheck 2018. NVIDIA GeForce GTX 1060 (Desktop) vs Qualcomm Adreno 540. Date of retrieval 12.11.2018. <https://www.notebookcheck.net/GeForce-GTX-1060-Desktop-vs-Adreno-540-7336-7714.247598.0.html>
19. Nvidia Corporation. GeForce GTX 1060. Date of retrieval 12.11.2018. <https://www.nvidia.com/fi-fi/geforce/products/10series/geforce-gtx-1060/>
20. Nvidia. Shop GeForce graphics cards. Date of retrieval 15.1.2019. <https://www.nvidia.com/en-us/shop/geforce/?page=1&limit=9&locale=en-us&gpu=GTX%201060&gpu-filter=RTX%202080%20Ti~7,RTX%202080~29,RTX%202070~20,GTX%201080~1,GTX%201070%20Ti~9,GTX%201070~10,GTX%201060~28,GTX%201050%20Ti~17,GTX%201050~23&sorting=lp>
21. Nvidia. Virtual Reality Technology. Date of retrieval 15.1.2019. <https://www.nvidia.in/object/vr-system-requirements-in.html>
22. Riftcat. Adventure awaits you!. Date of retrieval 21.1.2019. <https://riftcat.com/vridge>
23. Robertson, Adi 2018. Microsoft says it's no longer planning VR support on Xbox. Date of retrieval 28.11.2018. <https://www.theverge.com/2018/6/20/17485852/microsoft-xbox-one-no-vr-headset-support-windows-mixed-reality-e3-2018>
24. Samsung Corporation. Gear VR with Controller. Date of retrieval 10.11.2018 <https://www.samsung.com/global/galaxy/gear-vr/>
25. Samsung Electronics Co., LTD. Gear VR with Controller. Date of retrieval 29.11.2018. <https://www.samsung.com/global/galaxy/gear-vr/>
26. Seibert, William 2017. Virtual Reality Then: A Look Back at the Nintendo Virtual Boy. Date of retrieval 10.11.2018 <https://www.techspot.com/article/1085-nintendo-virtual-boy/>
27. Sony Corporation. PlayStation VR. Date of retrieval 10.11.2018 <https://www.playstation.com/fi-fi/explore/playstation-vr/>

28. Statista 2018. Global unit sales of current generation video game consoles from 2008 to 2017 (in million units). Date of retrieval 28.11.2018. <https://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/>
29. Statista. Global VR headset unit shipments market share by vendor in 2016. Date of retrieval 15.1.2019. <https://www.statista.com/statistics/755527/global-vr-headset-market-share-by-vendor/>
30. Steam. Virtuaalitodellisuus Steamissä. Date of retrieval 19.1.2019. <https://store.steampowered.com/vr/>
31. Unity Technologies 2018. Google VR hardware and software requirements. Date of retrieval 28.11.2018. https://docs.unity3d.com/Manual/googlevr_requirements.html
32. Unity, 2016. Building your Unity game to an iOS device for testing. Date of retrieval 7.1.2019. <https://unity3d.com/learn/tutorials/topics/mobile-touch/building-your-unity-game-ios-device-testing>
33. Unity. Optimizing graphics performance. Date of retrieval 21.1.2019. <https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>
34. Unity. Public Relations. Date of retrieval 26.1.2019. <https://unity3d.com/public-relations>
35. Zak, Robert 2017. How to Connect a PS4 Controller to Your Android Device. Date of retrieval 15.12.2018. <https://www.maketecheasier.com/connect-ps4-controller-to-android/>

```
void AttachObject()
{
    if (Input.GetTouch(0).phase == TouchPhase.Began)
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit, float.MaxValue, 1 << 10))
        {
            if(hit.transform.gameObject.GetComponent<Attachable>() != null)
            {
                hit.transform.SendMessage("OnAttach", currentItem);
                currentItem = null;
            }
        }
    }
}
```