

Augmented Reality app that educates the user about the lunar surface

By

Jonathan Roddy

Student ID: K00212118

Course: LC 233 Software Development

Supervisor: Carol Rainsford

Date: January 2021

**Table of Contents**

[1 Introduction 3](#_Toc61204403)

[2 Android development with Augmented Reality 4](#_Toc61204404)

[2.1 Introduction 4](#_Toc61204405)

[2.2 Technologies required. 4](#_Toc61204406)

[2.2.1 Hardware Development Environment: 4](#_Toc61204407)

[2.2.2 Software Development requirements 5](#_Toc61204408)

[2.2.3 Support requirements: 5](#_Toc61204409)

[2.3 Android Integrated Development Environment’s (IDE) 5](#_Toc61204410)

[2.3.1 What is an IDE? 5](#_Toc61204411)

[2.3.2 Android IDEs 7](#_Toc61204412)

[2.4 Augmented Reality Application Programming Interface (AR API) 11](#_Toc61204413)

[2.4.1 What is an API? 11](#_Toc61204414)

[2.4.2 ARCore 12](#_Toc61204415)

[2.4.3 Vuforia 12](#_Toc61204416)

[2.5 3D Modelling 13](#_Toc61204417)

[2.5.1 Blender 13](#_Toc61204418)

[2.6 Database 13](#_Toc61204419)

[2.6.1 Firebase 14](#_Toc61204420)

[2.7 Text to Speech 14](#_Toc61204421)

[2.8 Technology Conclusion 15](#_Toc61204422)

[3 Educating users about Moon via Augmented Reality 17](#_Toc61204423)

[3.1 Introduction 17](#_Toc61204424)

[3.2 The Moon 17](#_Toc61204425)

[3.3 Similar Apps available 18](#_Toc61204426)

[3.4 Resources Required 19](#_Toc61204427)

[3.5 Data Collected 20](#_Toc61204428)

[3.6 Effective edutainment 20](#_Toc61204429)

[4 Conclusion 23](#_Toc61204430)

[5 Bibliography 24](#_Toc61204431)

**Table of Figures**

[**Figure 1:** Unity Game Engine (AXON, 2016) 8](#_Toc61204432)

[**Figure 2:** Visual Studio (Microsoft, 2019) 9](#_Toc61204433)

[**Figure 3:** Android Studio Editor (Williams A. , 2019) 10](#_Toc61204434)

[**Figure 4:**Amazon Sumerian Console (Skarredghost, 2018) 11](#_Toc61204435)

[**Figure 5:** Android IDE Comparison 16](#_Toc61204436)

[**Figure 6:** UI Positions (Dori, 2019) 23](#_Toc61204437)

# Introduction

In this research paper, I aim to continue to explore all aspects needed for my final year project further beyond what I have covered in my literature review. In my literature review, I have researched 2 main technologies that I will be implemented for my project: Augmented Reality (AR) and Text-to-Speech (TTS). Having done this, it allowed me to be able to familiarize myself with both technologies so that I can be confident enough in utilizing them. I learned what they are, how they started, what is evolved when implementing them, and the components that are made from them.

This paper's goal is to highlight the technical knowledge required as well as the theme of the project. That includes aspects like Android Development, implementation of Augmented Reality, integrating Text-to-Speech, and being able to teach users interesting knowledge about our Moon. I will be researching each of those in-depth so that I may be able to be confident in my knowledge and my technical preparation before heading into the development stage.

There are several ways to complete a respectable Android application and I indent in finding the best solution that will both benefit this project as well as improving my studies and skills as a Software Developer. The needs for my technical research would include an Integrated Development Environment (IDE) for developing the software for the application, the correct Augmented Reality Application Programming Interface (AR-API) to utilize, what 3D rendering software is needed as well as a suitable database storage solution. Once I have researched and found the correct architecture infrastructure, I can continue with researching the theme. At the end of this project, I want to be able to teach users about the Moon, but I do not have any experience of teaching users over an application. Once I am happy that I have enough information I need, I will continue this projects life cycle into its design and implementation section.

# Android development with Augmented Reality

## Introduction

In this chapter, I will delve into what technical knowledge and skills are needed to be able to develop an Android application that implements an Augmented Reality (AR) Application Programming Interface (API). I have not developed an android application before I started this project so this will be my first time developing an application. This brings new challenges and opportunities to which I can better myself with these new technologies and techniques. Such as, developing on an android integrated development environment, utilizing an AR API and TTS, 3D modelling and rendering as well as communication with a cloud database.

## Technologies required.

To be able to develop my final year project successfully I would need a mixture of hardware and software requirements. From my investigation in my literature review, I know what is required to build an Augmented Reality (AR) application. I have delved into the area of AR so I now can deduce what hardware components are required. I have listed below under each section what I need in preparation for this application. The list of technologies is as follows:

### Hardware Development Environment:

* Desktop computer: for the development of the application
  + Processor (CPU)
  + Graphics card (GPU)
  + Random Access Memory (RAM)
  + Motherboard
  + Power Supply Unit (PSU)
  + Input/Output peripherals
    - Mouse
    - Keyboard
    - Monitor
* Internet connectivity: to be able to connect to the internet for research and references.
  + Stable bandwidth connection
* Android Smartphone: for the implementation of the application
  + Processer
  + RAM
  + Integrated Graphics
  + Battery
  + Rear Camera
  + Screen (LCD or AMOLED)
  + Gyroscopes
  + Touch screen(digitizer)
  + Wi-Fi receiver
  + Android version 8.1 (API 27) or later.

### Software Development requirements

* Windows 10 (Operating System to house all applications)
* Android IDE (Programming the code)
* AR API (Import the AR libraries)
* TTS API (Read Text aloud to the end-user)
* Firebase (Google cloud Database)
* Blender (3D Model creation)
* Scene Viewer (3D Rendering)

### Support requirements:

* Supervisor meetings with L.I.T. given me their professional guidance and experience as needed.

Now I would like to talk about each component further explaining to you what they are as well as some common tools available.

## Android Integrated Development Environment’s (IDE)

### What is an IDE?

An IDE can be defined as a developer’s toolkit. When you deal with a programming language, having a good IDE will make all the difference. It can help you prevent mistakes that would not otherwise be caught until it is time to compile or test your code. A good IDE has several essential features that it gives to the user which helps productivity move smoothly.

**Code editor:**

Most of the time is spent typing text into an IDE. To type your code, an IDE must have a place where you are comfortable to use you. If the editor is inefficient or clunky, then you will not want to spend your day using it. A good IDE even has editor emulation modes so that the same key combos that have already committed to muscle memory.

**Syntax highlighting**

An IDE display keywords from the language you are using in one colour, variables in another, libraries, and functions in another, etc. This emphasis is a small addition to your life in coding, but one that gives context to what would otherwise be a flat monotone text wall.

**Code Linting**

An IDE highlights your errors for you if you make a mistake while typing your code. This automated checking of code is known as ‘linting’.

**Code correction**

Linting is the error catching process and based on a rudimentary schema of expected syntax, many IDEs can do that. Although a brilliant IDE implies error correction. Best still, sensible corrections resulting from the way the programming language operates in real life should be proposed by an IDE. In other words, improvements that make sense for Java, not C# or Python, are recommended by a Java IDE Of course, it is not all perfect. Only a fix based on following a language standard can be suggested by an IDE, and in the wrong sense, that can ruin your project. It is important to have your interpretation of the terminology and what you are attempting to do but auto-corrections are useful for detecting apparent mistakes.

**Project awareness**

Code and assets are packed together with either physically or digitally so that you cannot leave any critical elements behind until you are about to deliver the code. The configuration of your code and its libraries is abstracted by certain IDEs such that you can see anything on which your code depends, both the system-wide toolkits you built by default (but your users do not and the custom code you built locally.

What amount of granularity you do require depends on your management abilities, the monitoring of dependencies, and the vocabulary and libraries you are using? But no matter how complicated or basic your project is when looking for archives, libraries, and properties, it is easy for your IDE to know which directory to default to.

**Environment awareness**

Typically, an IDE has a management framework to support you customize which code libraries to use the runtime, compiler choices, and to regulate version control. On any device, this is important, but on a production computer with many iterations of tools mounted, it is critical. For eg., it is not uncommon to install both Qt4 and Qt5 toolkits, Python 2 and 3, various Java runtime versions, 32-bit and 64-bit libraries, GCC and LLVM, and so on. It is good to have an IDE to monitor how the idea is designed and bundled for predictable performance.

Another essential convenience feature is designing, running, and debugging your code from inside your IDE. Of course, to also run your compiler or launch a debugger, you do not need your code-editing application, but having those features in the same interface brings unity to a process that is often different.

Now that we have a quick understanding of what a great IDE is, I will research and document a few IDE’s that can be used for android development. I will investigate each one to make sure they follow best practices and based on the use case of my application what best suits my needs for this project.

### Android IDEs

During my time as a student of software development, I have encountered several IDE’s both for my academic course as well as my personal development. I have experience with both good and bad IDE’s so I will use my intuition, the segments talked about in section 2.3.1 above, as well as the use cases in which the application will be used.

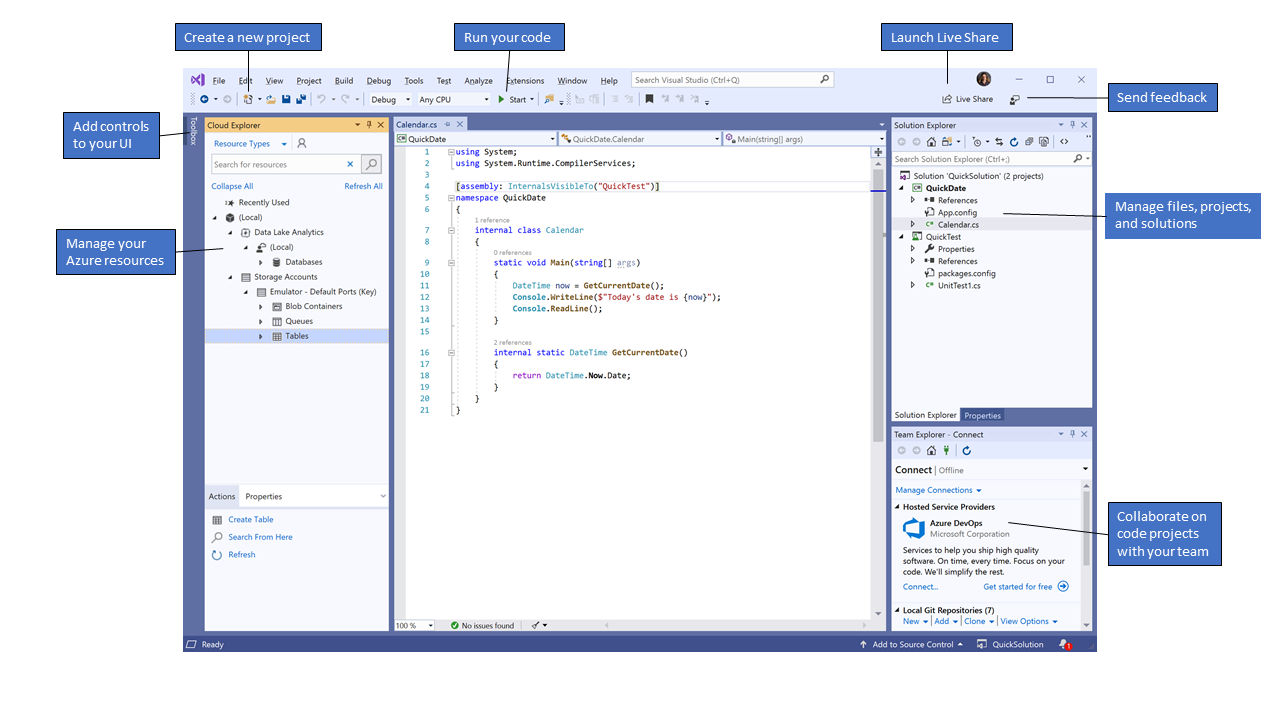
**Unity**

Unity is, as of 2018 a cross-platform game engine developed and owned by Unity Software Inc, (doing business as Unity Technologies). This game engine can be used in several ways to create, render, and code assets in three-dimensional, two-dimensional, virtual reality and augmented reality as well as simulations (AXON, 2016). However, this game-engine was first developed to be used for the gaming industries, it is now not limited to this. Its constant growth and evolvement each year has allowed it to be used in the film. Automotive, architecture, engineering, and construction. Unity Technologies offers a wide range of support and assets to its users and developers. They offer a program called Unity Learn which is an online learning course with more than 750 hours of on-demand content for all skill levels (Unity Tecnologies, 2021). Unity is free to download software that can be installed in a matter of a few minutes. It can be downloaded for Windows, Linux, and MAC and can be used for up to 25+ different platforms. Eg. IOS, Android, Windows, WebGL, PS4, Xbox, OcalusVR, Android TV, tvOS, Nintendo Switch, ARCore, Microsoft HoloLens, and Magic leap. However, you must create a free trial account to be able to develop and explore the ecosystem. If you wish to release anything or get full support, you must create a subscription-based on the user’s needs. Unity offers a wide range of licensing for its different solutions.  
Unity Mars one of many software solutions released by Unity but its sole function is to build powerful intelligent AR applications. It can compile real-world data in real-time on the Unity editor while being able to customize and build to any AR ready platform. This does not come for free as the annual license is $600. (Unity, 2021)

**Figure 1:** Unity Game Engine (AXON, 2016)

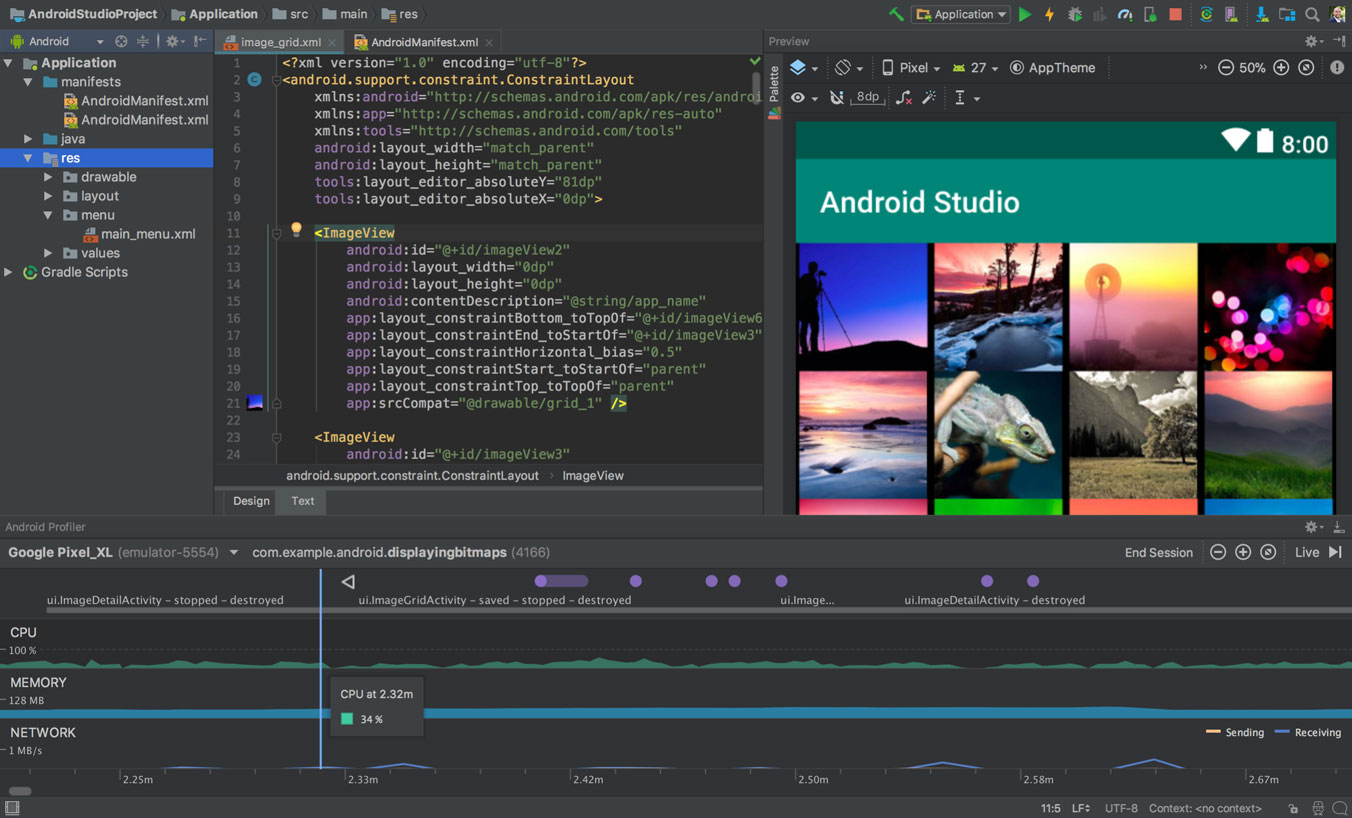
**Microsoft Visual Studio**

Visual Studio is an IDE created by Microsoft. It is a creative launching pad that can be utilized to edit, debug, build, and then publish an application. It is commonly used to develop programs for computers but can be utilized for websites, web apps, web services, and mobile apps. The most popular productivity features offered by the visual studio are as follows: squiggles and Quick Actions, Code Clean-up, Refactoring, Visual Studio Search, Live Share, Call Hierarchy, CodeLens, Go to Definition and Peek Definition (Microsoft, 2019). All these features are added together to ensure that visual studios perform as a great IDE to use for any developer and their projects. The Visual Studio ecosystem allows the developers to install plugins and API’s as they need them instead of having to install everything all at once (Visual Studio 2019, 2019).   
Visual Studio is a free download that can be downloaded to Windows while a separate version can be found for macOS. This free version will allow you full access to the Visual Studio ecosystem. During my educational studies, I had access to Visual studios community edition and was using it for three years allowing me to get familiar with its features and UI.

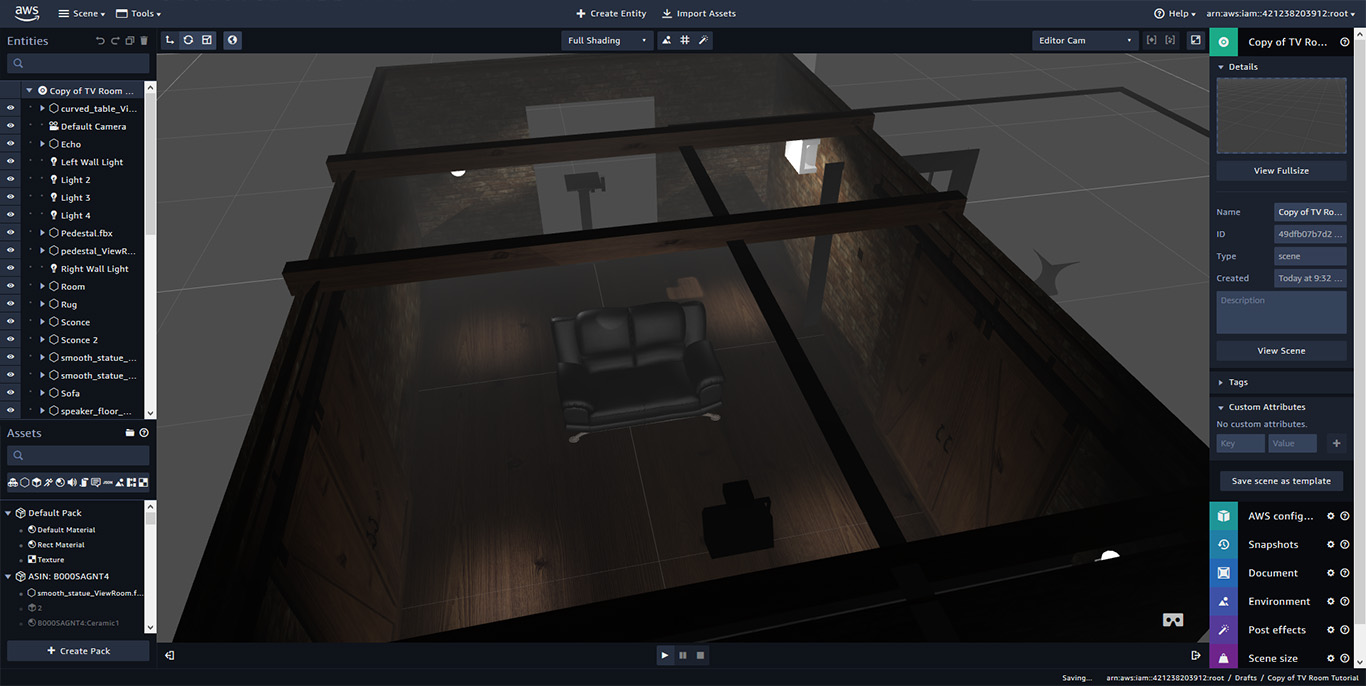
**Figure 2:** Visual Studio (Microsoft, 2019)

**Android studio**

Android Studio is Google's official IDE-based to develop native android applications on the Android operating system. Android is built upon a Linux distribution, so it is an open-source operating system. Android studio works on the IntelliJ IDEA and IntelliJ IDEA Community Edition which both are open-sourced and free to anyone. IntelliJ IDEA is just an IDE that was built on Java. Google has decided that they will provide the IDE and all its tools for free which will allow a lot more developers to use this. Google Inc owns and distributes Android OS so having their IDE makes sense. It can be downloaded for free on Windows, Linux, and macOS and can be downloaded via a subscription-based service (Google, 2021). Since its release, Android Studio has had Java as its main programming language but as of May 7th, 2019 Kotlin has become popular with Google themselves (Lardinois, 2019). Kotlin is an already existing programming language that is both powerful and simplistic to use. This is very appealing for any developer. If an application is to be released, then a once-off payment of $25 is required. I do not have prior experience before starting this project but completed a semester of mobile development using this giving me quite an insider of the IDE and its many functions.

**Figure 3:** Android Studio Editor (Williams A. , 2019)

**Amazon Sumerian**

Amazon launched Sumerian in November 2017 which is a game engine that runs completely on the web. This approach is different from most as it does not require you to download or install anything on your computer. This follows Amazon's Web Services approach to computing in cloud architecture. This allows developers to simply log into their console and start working on an AR or Virtual Reality (VR) environment without the need for a good pc. There is no IDE that is being used as the software itself is encapsulated into Amazon's web console. The only coding that can be uploaded or changed is JavaScript (Amazon, 2021).  
If you want complete freedom with the projects developed with Sumerian this cannot be done due to it being locked into the Amazon environment. The price of using this software solution depends on the needs of your application. There are two different types of pricing schemes you can apply for. 3D object storage and Scene Traffic. You are charged for what you use and no more, so the more complex the application the higher the fees. was lucky enough to have a quick meeting with one of the lecturers in Limerick Institute of Technology who has had hands-on experience using this software. He gave me great advice such as the limitations of the online IDE, use cases where this is better over other ones, complexity, and fees. Once I gave him my project information, he advised me that though this is a good online IDE it will not benefit me to use this due to its limitations and functions. I found a great tutorial on the implementation of Sumerian for a basic set up which the author gave great examples of uses and restrictions on this software (Skarredghost, 2018). From reading this and from comparing my implementation research this solution is ideal for this project.

**Figure 4:**Amazon Sumerian Console (Skarredghost, 2018)

## Augmented Reality Application Programming Interface (AR API)

Now that we have gone through some of the most common Android IDE’s on the market that is being used for AR development, I would like to take some time to talk about AR Application Programming Interface. I need to use an API in my development because of its fast superior functions and features available. In my literature review paper in Chapter 2, I have discussed what AR is, the main components needed, where it is being used today, and security. I will use the knowledge gathered here to help me find the best AR API that is available to me. There are a few AR APIs out there that have different approaches, so it is important I find the correct one for this project. I have picked the top two AR APIs that are used within Android development to investigate as they all offer great aid while also being unique to one another.

### What is an API?

Typically, an API is a series of protocols, routines that aid the developer to complete a task. An API also defines how two or more programs communicate with one another. There are loads of free or paid API’s that one can use, and they usually do not need more than a few lines of code to connect into. With a connection established an API allows a developer to have extra features or functions that they did not program themselves. It can also add complex mechanisms to extend functionality (Fisher, 1988). They can be found in all operating systems, websites, and applications. Thousands of APIs are used daily with new additions being made available with every new update or software release. API’s have one of three purposes: System API, Process APIs, and Experience APIs. The job of a System API is to access and maintain some data while also managing the configurations of a system. A Process API then takes the take from the System API and incorporates it to create a new view. The Experience APIs add substance to both the System and Process APIs (Norton, 2019).

With each type of API, there are complex architectures and implementations in place but for this research paper, I am only interested in APIs that are for AR. Having an AR API allows me to import libraries and functionality already set out by the community to aid me in my development faster. I have combined the following two APIs and will compare each of them.

### ARCore

ARCore is Google's platform that they have developed and was released on the 1st of March 2018. Its sole purpose is to aid in the development of augmented reality. This software allows the user's phone to understand its environment and capture its surroundings. It is made up out of three key components, Motion tracking, Environmental Understanding, and Light estimation. Motion tracking works by using the cameras on the back of the smartphone to its advantage. It scans the area and makes a 3D representation of its surroundings. It will ask the user to slowly tilt the phone left to right and will pick certain markers as a reference point. ARCore uses Simultaneous Localization and Mapping (SLAM) to understand the position in which you are standing. Environmental Understanding is where the reference points are used to determine a flat area such as a floor or tabletop. This allows the 3D models to be placed on that area for the user to see. Light Estimation is where ARCore detects light from what it can detect and try to mimic that on the 3D models for more of a realistic representation. ARCore will work on most android phones that have Android 8.1 (API 27) or later (Google, 2020). They have a full list of supported devices on their website for anyone to check. It however cannot be used on any other smartphone outside of native Android. It can be implemented in Android Studio, Unity, Unreal Engine, and IOS. Implementing it comes for free and only takes a few lines of code for it to be fully supported and working.

### Vuforia

Vuforia is another such API that offers great AR aid. Vuforia can only be used with the Unity IDE so it limits itself with other IDE options. The features are very similar to what ARCore has to offer but it is not as advance as it. It too has Motion tracking, environmental understanding, and light detection but ARCore is much more adapted to knowing its surroundings and its environment in which it orientates (Dudkin, 2019). Once Vuforia is implemented with Unity it can be used on any AR-ready device not only on Android. This allows a further reach in comparison to ARCore. Developing with Vuforia is free but if you want to publish your application then you will need to buy a license.

From my project, I can tell that ARCore offers me a better outcome as its features are more adapted and advance than that of Vuforia. I do not mind being restricting to only Android devices as this project was solely aimed at Android devices in the first place.

## 3D Modelling

Now that I have investigated IDEs and AR APIs the next area is the 3D model aspect of the application. I have not made any models before for an application, but I have studied AutoCAD during my secondary level education. AutoCAD is used to sketch up and design 3D models with the help of 2D drawings. This experience should help me develop my models for this project such as the Moon, Earth, and spacecraft. 3D models are vital for AR as that is the models that the user will be interacting with via their smartphones. These 3D models can be anything if they are readable with whatever AR API that it is being called by the IDE.

### Blender

Blender is the most common and popular of all 3D rendering software as it is free, and it is open source. It can be downloaded on Windows, Linux, and macOS. Not only does it allow users to create 3D models, but it also supports the entirety of the 3d pipeline. This means that it can be used for such things as rigging, animation, simulation, rendering, compositing and motion tracking, video editing, and 2d animation pipeline (Blender, 2021). There is a vast community that has grown together as blender did with each update. They provide helpful videos and training to get up to scratch with everything you might need to be able to start developing with Blender. 3D models that are created inside Blender are not restrained to any software system as it can be exported as different object files as well as being able to import other files. You do however would need a strong enough computer to be able to work effortlessly as the system itself is quick heavy to use on its own. It uses the power of the computer's graphics card to be able to render the image, scene, or model in the highest quality that the user has given. Utilizing Blender for my project is a must as the features and functions it provides are next to none.

## Database

Having a database nowadays is a must for any application that can connect to the internet. The ability to store data online that can be accessed anywhere, and at any time is beneficial beyond belief. It allows applications to be less heavy on storage as no files or data are being saved locally. If a database is set up correctly following best practices it should be secure, reliable, and efficient. For my project, I will be utilizing a database as I want the product to be lightweight on each device. For me to do this I will need to implement a database that can store 3d objects, data in plain text, and coordinates on each model. From researching my needs one such database solution has popped up and that is Firebase.

### Firebase

Firebase is an online real-time no-SQL database that has been developed by Google for mobile applications as well as web applications. A no-SQL database differs from an SQL difference in the ways that there are no relationships set up between tables or columns. Each table is created as a JSON object for faster more effective data transmission. Firebase comes free to use and can be set up online via a web browser or implementing via Android Studio IDE. There is a free plan that has no restrictions on any features but once a certain threshold has been passed then it will charge the user. What it offers is authentication. Cloud Fire store, Cloud Functions, Hosting, Firebase Machine Learning (ML), Realtime Database, Storage, Test Lab, and Google Cloud Platform (Google, 2021). If the needs are more complex, then Google will charge you monthly for what you need and nothing more. Android Studio can implement most of the features of Firebase insides its own IDE as Google owns both ecosystems.

## Text to Speech

One of my objectives for this application is to enable the use of Text to Speech (TTS). In my literature review paper, under chapter 3, I have talked about what TTS is, the main components that makeup TTs as well as the main components that make up the technologies. Here I would like to talk about how I would implement this in my project.

There are many ways in which I can achieve having TTS in an Android application like having a pre-recorded narrator read outlines of text or having a computer synthesize the text in real-time. Having an actual human narrator would have been the way to go if not for the growth of TTS technology. Nowadays it is a lot easier for an API to do the talking as not only is it much cheaper to run but the synthesized voice can speak in many languages. When decided on the correct implementation of TTS I will look at each of my potential IDEs to see if they have pre-existing libraries. For both Unity and Android studio there is indeed a library that a developer can import that has all the necessary features to run a full TTS engine. Visual Studio has a separate System. Speech only works on the .NET framework. All my IDEs have TTS functionality, but Unity and Android Studio share the same library. Google has complete documentation on how to use it as well as how to manipulate it within Android Studio itself. Unity is relying on plugins and external documentation to help developers use this library.

## Technology Conclusion

I have now talked about IDEs, AR APIs, 3D Rendering, TTS, and databases that I believe could be used to develop and implement this project. Below in figure 5, I have captured some data and compared them with each of the other IDEs to help aid me in the decision of the IDE that will be used. Seeing as I will be aiming for this project of mine to be developed for Android, I believe that Android studio is more than capable of what I need. It can be downloaded to my workstation for free, it comes with great online support and documentation and can be used fluidity with ARCore and Firebase. Having my project encapsulated within the Google ecosystem not only gives me great resources, but they have also been developed to be used together with little to no resistance. So, I will be using Android Studio for my android development IDE with the use of the TextToSpeech engine, ARCore for my AR API implementation, Blender for my 3D modelling and rendering, and storing it all on Firebase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topics** | **Unity** | **Visual Studio** | **Android Studio** | **Amazon Sumerian** |
| Development Platform | Windows Linux macOS | Windows macOS | Windows Linux macOS | Browser |
| Price/Licensing | Unity Enterprise: 10 Person Team @ $200/month = $2,000/month Unity Pro: 1 person / annual plan = $1,800 Unity Plus: 1 person / annual plan = $480 Unity MARS: 1 person / annual plan = $600 per seat Unity Free trial: 30 days | Community edition: Free Business: $45/month Enterprise: $250/month | Free release app to PlayStore: $25 | Free trial: 50Mb with 100views(5Gb) for one year Scene Storage: $0.06 per GB/month Scene Traffic: $0.38/month |
| Languages | C#, Boo, JavaScript, Unity Script | C, C++, C++/CLI,  Visual Basic, .NET,  C#,  F#,  JavaScript,  TypeScript, XML,  XSLT,  HTML,  CSS, Python,  Ruby, Node.js,  M | Java, Kotlin, C++,  C#,  Xamarin,  LUA,  JavaScript | Sumerian editor JavaScript HTML |
| AR API | Vuforia | ARKit | ARCore | Sumerian |
| Personal Experience | None | Intermediate | Basic | None |
| Notes | Great IDE Pricing is too much for me right now Trial period too short for learning and implementing on | Great IDE Great choices of programming languages Most comfortable using Would need professional version to release to public | Great IDE Nice choices of programming language to choose Fits perfectly into the Google ecosystem Free to use with all features available Locked into only Android devices Open Source | Basic IDE Expensive to run Locked into Amazons Ecosystem Little to no coding required Basic in functionality |

**Figure 5:** Android IDE Comparison

# Educating users about Moon via Augmented Reality

## Introduction

For my second chapter, I will talk about the theme of my application, educating users about the Moon. This project will utilize AR and Text Speech technologies to aid in education about the Moon. During my literature review, the word ‘edutainment’ came up which bests describe this project I am about to undertake. Under section **2.7** of my literature review, I talk about the components that make up a good edutainment environment. Edutainment is a process in which you are entertaining people while also teaching them something new that they did not know before. This can be done via software, programs, or television (Cambridge Dictionary, 2021). There are plenty of edutainment programs and television shows that not only entertain their audience but also educates them as well. The top three popular edutainment shows online that flearningstudios.com has rated are as follows, The Mind Explained Series by Vox, which uses abstract art over real-life footage, CrashCourse which is an animated YouTube series and Ted-Ed which is a subsection of TED the talk seminars.

I choose to develop an application based around the Moon as I find myself fascinated with Astronomy and with the Moon. I would love to use an application like what I will develop to help my knowledge grow further. For this to happen I need to be able to: Understand how to educate users effectively and efficiently, what assets will I show to users, what assets about the moon do I need, a timeline of previous, current, and future space missions, etc. I must portray this information in the best way possible to ensure the most information is displayed.

## The Moon

First off, we need to describe what a moon is. A moon is described to be any celestial body that orbits around a planet or dwarf planet. A moon cannot be described as an object that orbits its parent star however it does so by orbiting around its planet. The name ‘Moon’ was first given to the Earth's Moon way before the realization of other planets moon. It is the only celestial body that has the same name as the defining characteristic that describes it. We do not know how, when, or why Earth got its Moon, but it has been around before life itself. There is 3 hypothesis that has been agreed that could explain how we got out the moon. I will explain these when gathering my facts later in section **3.5.**

The Moon has been visible to us for our entire human civilization and longer. It has inspired religions, art, culture, and music throughout the years down to its mysterious nature. For centuries people believed that Earth's Moon was one of a kind, but thanks to the progressive improvement of technologies, telescopes allowed us to see beyond what was possible. It has inspired me enough to want to do my final year project around it. I will gather as many interesting public facts and knowledge that has been gathered collectively by governmental organizations to use for this project. There have been several space missions to the moon as well as ones that survey it. Man has even landed on the moon several times but not since 1972 on the Apollo 17 mission. There are plans that NASA will have men back on the moon in 2024.

Without the involvement of both the Sun and the Moon, Earth would not have the ability for its tides to go in and out as it has been. It is tidally locked to our planet which means that only one side of the moon can ever be seen from Earth. There is so much that we as humans know about the moon but there is far more that we do not. Without it, the world we know of today would not have been able to thrive as we did.

So now let us look at what apps are out in the public right now that compares to my educational AR app and see what they do. Researching other application is important as it allows me to make sure what my plan is for my project is unique.

## Similar Apps available

Let us talk about similar apps that let the user explore the Moon. I am only looking at the Android Play Store for reference as my application will not be for IOS. The applications I will be looking at must have the following similarities: must be an Android application, uses AR to help display models of the Moon, must be free to download, and must be easy to use.

* **Giant Moon AR:**

This application is simple but fun. It allows the user to point their phones to the sky and it will display an overall enlarged version of the Moon using AR. The user can then see what this new oversized Moon would like in real-time.

* **Apollo’s Moon-Shot AR:**

This application has many more features that it gives to the user. It is based on the Apollo space missions that NASA has conducted and teaches users about the men, spacecraft, and tools needed. It has videos showing the different missions, the journey to the moon, previous space missions that lead to the Apollo program as well as much more. It AR in a few exciting ways, it shows models of the spacecraft, the rockets flying out of the atmosphere, the moon itself and you can take selfies as if you were on the moon itself.

* **Moon AR:**

This AR app displays a model of the moon to the user showing a small window of facts at the bottom of the screen. If the user moves towards the Moon you can see a small Apollo spacecraft that orbits it.

* **Moon Locator 2021:**

This application gives the user a defined path in which the moon itself will show depending on the user’s geolocation coordinates. It uses AR to show on the screen what path in the sky the moon will take.

* **Moon Atlas 3D:**

This application is not an AR one, but it does have a similar idea. It shows a High Definition (HD) model of the Moon with an overlayed atlas on top. The user can zoom in and out from within the model to be able to see all the meteor's holes and landing spots.

* **Missions to the Moon:**

This application uses AR within it, but it is marker-based meaning it uses a picture that is on a flat surface to display its models. This application is part of a scientific book that has the same name as the app. It overlays 3d models over some items and plays videos in some image boxes.

All these applications are fun and educational to use. They all have one main objective and do it well. There are a lot of other similar applications, but they do not either use AR, not free, or available on the Android PlayStore. None of the above applications have a defined learning objective nor do any of them use Text to Speech.

Now that I have talked a little about the Moon, and applications that are similar I will now talk about the resources that I need to build my application.

## Resources Required

For me to be able to build a successful application that teaches users about our Moon I need to be able to make sure the facts I tell are 100% and are credit correctly. None of the information that protrudes should be fake nor made up. I will gather as much as I can, and I will narrow down the selection to build the project in time as this project is limited due to its time. I will use the governmental website as much as I can over normal websites with little to no reputation.

Moon:

* I need accurate images and topography of the lunar surface. NASA has released HD images of the Moon so that they can create an accurate 3D model and I intend to use them.
* I need a geologic map of the moon that highlights different sections of the Moon. NASA can provide this, and it is for the public to use.
* I need a complete timeline of all manned, unmanned missions to the Moon. NASA has built a timeline of all missions ever since 1959 to the present and the future.
* I need a Moon phrase calendar API that is accurate for 2021.
* I need a list of interesting facts that can be shown to the user.
* I need data and images to show the user what a Red Moon is.

**Earth:**

* I need accurate images of the Earth to be able to build an accurate model.

## Data Collected

Please see Bibliography for a complete reference of the below assets. I have gathered resources that are accurate in its data below.

* Colour and displacement assets of the Moon (Wright, 2019).
* Geological Map of the Moon (Corey M. Fortezzo (USGS), 2020).
* Timeline of all missions to the Moon (Williams D. D., 2020)
* Moon Phases API (wdisseny, 2021)
* Interesting facts about the Moon (Space Facts, 2019)
* Red Moon data (Kher, 2021).
* Texture image of the Earth (Shahrai, 2020).

With the resources and data above, I will be able to use this information to build my application that will be accurate and truthful throughout. I can use these assets with Blender to be able to build realistic models that I can use within Android Studio.

## Effective edutainment

To be able to portray the data that I have gathered above in an effective edutainment environment, I need to follow best practices that I have eluded too in my literature review. Under section **2.7** I learned that you need to pull together three different areas to build an effective learning environment. When developing I need to pull aspects from educators, game design, and multimedia. I will need to keep these into consideration when continuing with my design and implementation section of this project. I want to use the best user interface (UI) and user experience (UX) to aid me throughout the development stage.

When trying to teach someone something new you need the user to not lose interest fast as they will not be able to hold any of the information. Kevin Daum, a successful entrepreneur has given out a five-step process in which you can teach anyone anything (Daum, 2013). His five steps are as follows:

1. **Create a clear curriculum -** Have structured content with clear objectives and milestones.
2. **Make the material matter -** Create detailed documents, videos, and pictures.
3. **Present with purpose and passion -** Be enthusiastic about the subject and energized.
4. **Let the learners lead the learning -** Allow users to have their exercises that encourage further questions.
5. **Reinforce with repetition and response -** People tend to learn more from their failures so provide frequent specific feedback.

When talking about the best practices someone needs to be able to effectively use multimedia, I have found 12 principles that Mayer and Moreno have developed. These principals were developed to aid people with digital learning environments (Mayer, 2009). They are as follows.

1. **Coherence Principle** – Remove any extraneous words, pictures, or sounds.
2. **Signalling Principle** – Add cues that highlight the material.
3. **Redundancy Principle** – Using graphics and narration together rather than pictures and plain text.
4. **Spatial Contiguity Principle** – Place words and pictures close together instead of far apart.
5. **Temporal Contiguity Principle** – Place the words and pictures simultaneously instead of successively.
6. **Segmenting Principle** – Keep the lesson user-paced instead of continuous unit.
7. **Pre-training Principle** – Using names and characteristics that are previously known.
8. **Modality Principle** – Using graphics and narrations instead of animation and plain text.
9. **Multimedia Principle** – Use words and pictures instead of plain text alone.
10. **Personalization Principle** – Use of conversational style rather than formal style.
11. **Voice Principle** – Use a friendly voice instead of a robotic one without tone.
12. **Image Principle** – Not a need to have the narrator’s image on the screen.

These 5 educational steps and the 12 principles of multimedia will both contribute to my implementation. I will use these to my advance when designing the application. From the above, I can plan out a strategy to educate users while following all the principles to the best of my ability.

The last aspect I would like to look at would be game design. I have not made any form of game before, but I need to be able to do so for this project. They are many different types of games available between video and board games as well as the subcategories within them. For this application, I would like to investigate android game design and see what professionals advise. I do not aim to make my project a game, but I will investigate what helpful information they can offer me. Gamedev.net has a list of their best game design user experience best practices (Dori, 2019). This comes from a senior android game developer who has created several well-reclaimed apps.

1. **UI Positions** – Figure 6 shows a great representation of a good UI Position for both smartphones and tablets. Have all the important buttons close the easy side of the screen (Store button, more apps, ads), and anything you do not want the user to click have them out of reach (exit).
2. **Sliders** – Utilise sliders where possible to allow for more content to be visible to the user. This can allow the user to stay in one scene. If showing a lot of content make sure there is an item half in view so the user can see there is more to see.
3. **Pop-Ups** – Good for delivering abstract messages to the user. Make sure to have the background out of focus and transparent compared to the pop-up. If there is information you want the user to see, make the close button small or hard to see initially.
4. **User Choices** – Plan out what choices you want the user to take. Associate a positive colour with the choices you want and give a negative colour to those you do not. For example, having green buttons and red buttons.
5. **Rewarded videos** – Give users the choice to watch a video so that they can get an in-game reward for spending their time. Usually, this video will be ads. Try not to tell the user what reward they will get so their curiosity kicks in.
6. **Draggable Objects** – Utilise the ability to drag items as it the user a false sense of control. Make sure the dragged item can still be seen even though it will be covered by a finger.
7. **Rate us dialogs** – Prompt the user to give a positive review of the Play store after a while of using your application. With more positive reviews, the better your application will be represented by other future customers.
8. **Store Dialog** – The design of your application store should be important. Have any characters in-game appear facing the user, make sure there is always a sale on, so the user believes it is more worth it than is, update the content on the store regularly.



**Figure 6:** UI Positions (Dori, 2019)

# Conclusion

Developing an AR Android application takes a lot of components to fulfil, from hardware to software specifications. This research paper has allowed me to investigate aspects that I have not done so before. I had to take into consideration the options of all the technologies that I could use as well as the edutainment rules. I have never developed an android application before so giving some time to the technologies helped me out. From developing this research paper, I now have a much clearer picture of what needs to be done and how it can be accomplished. I now know what Android IDE best suits the use cases, what AR API to use, more information on the database solution as well the TTS engine. Developing an application that teaches its users about the Moon needed me to be able to not only need the resources that allow me to display but also to teach. Getting some insight on what edutainment is as well as the different components that make it up. I look forward to continuing this projecting as the next stage is designing the application from the ground up.

# Bibliography

Amazon. (2021, 01 04). *Amazon Sumerian*. Retrieved from aws.amazon.com: https://aws.amazon.com/sumerian/

AXON, S. (2016, 09 27). *Unity at 10: For better—or worse—game development has never been easier*. Retrieved from arstechnica.com: https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/

Blender. (2021, 01 04). *About Blender*. Retrieved from blender.org: https://www.blender.org/

Cambridge Dictionary. (2021, 01 04). *edutainment*. Retrieved from dictionary.cambridge.org: https://dictionary.cambridge.org/dictionary/english/edutainment

Corey M. Fortezzo (USGS), P. D. (2020, 05 19). *Unified Geologic Map of the Moon*. Retrieved from astrogeology.usgs.gov: https://astrogeology.usgs.gov/search/map/Moon/Geology/Unified\_Geologic\_Map\_of\_the\_Moon\_GIS

Daum, K. (2013, 09 06). *How to Teach Anything to Anyone*. Retrieved from inc.com: https://www.inc.com/kevin-daum/how-to-teach-anything-to-anyone.html

Dori, A. (2019, 09 30). *Game Design User Experience Best Practices // Ultimate Guide*. Retrieved from gamedev.net/: https://www.gamedev.net/articles/game-design/ux-for-games/game-design-user-experience-best-practices-ultimate-guide-r5204/

Dudkin, I. (2019, 01 09). *VUFORIA VS ARKIT VS ARCORE: CHOOSING AN AUGMENTED REALITY SDK*. Retrieved from skywell.software: https://skywell.software/blog/vuforia-vs-arkit-vs-arcore-choosing-an-augmented-reality-sdk/

Fisher, S. (1988). *OS/2 EE to Get 3270 Interface Early.* InfoWorld Media Group, Inc.

Google. (2020, 12 12). *ARCore*. Retrieved from developers.google.com: https://developers.google.com/ar

Google. (2021, 01 04). *Android Studio*. Retrieved from developer.android.com: https://developer.android.com/studio/

Google. (2021, 01 04). *Firebase*. Retrieved from firebase.google.com: https://firebase.google.com/

Kher, A. (2021, 01 05). *Why Does the Moon Turn Red?* Retrieved from timeanddate.com: https://www.timeanddate.com/eclipse/why-does-moon-look-red-lunar-eclipse.html

Lardinois, F. (2019, 05 07). *Kotlin is now Google’s preferred language for Android app development*. Retrieved from techcrunch.com/: https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/?guccounter=1

Mayer, R. E. (2009, 10 21). *Multimedia Learning.* Cambridge University Press. Retrieved from edutopia.org: https://www.edutopia.org/blog/digital-spaces-12-best-practices-michelle-manno

Microsoft. (2019, 03 19). *Welcome to the Visual Studio IDE*. Retrieved from docs.microsoft.com: https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019

Norton, K. (2019). *API Definition & Meaning*. Retrieved from webopedia.com: https://www.webopedia.com/definitions/api/

Shahrai, A. (2020, 12 12). *High resolution satellite image of Earth in equirectangular projection.* Retrieved from shutterstock: https://www.shutterstock.com/image-illustration/high-resolution-satellite-image-earth-equirectangular-784406950

Skarredghost. (2018, 02 27). *Amazon Sumerian: what is it and hands-on review*. Retrieved from https://skarredghost.com/: https://skarredghost.com/2018/02/27/amazon-sumerian-hands-review/

Space Facts. (2019, 12 12). *Moon Facts*. Retrieved from space-facts.com: https://space-facts.com/the-moon/

Unity. (2021, 01 04). *Unity Mars*. Retrieved from Unity.com: https://unity.com/products/unity-mars

Unity Tecnologies. (2021, 01 04). *Everything you need to succeed*. Retrieved 01 04, 2021, from unity.com: https://unity.com/

*Visual Studio 2019*. (2019, 03 19). Retrieved from https://visualstudio.microsoft.com/: https://visualstudio.microsoft.com/vs/

wdisseny. (2021, 01 05). *Moon Phases API*. Retrieved from wdisseny.com/: http://www.wdisseny.com/lluna/?lang=en

Williams, A. (2019, 01 25). *Android Studio 3.3 advances with the Navigation Editor*. Retrieved from electronicsweekly.com: https://www.electronicsweekly.com/blogs/eyes-on-android/programming/android-studio-3-3-advances-navigation-editor-2019-01/

Williams, D. D. (2020, 12 28). *Lunar Exploration Timeline*. Retrieved from nssdc.gsfc.nasa.gov/: https://nssdc.gsfc.nasa.gov/planetary/lunar/lunartimeline.html

Wright, E. (2019, 09 06). *CGI Moon Kit*. Retrieved from svs.gsfc.nasa.gov: https://svs.gsfc.nasa.gov/4720