Android Game with Augmented Reality

Interim Report

University Of Leicester

Lukas Krasauskis

2021

Contents

[Aims and Objectives of the project 2](#_Toc88701634)

[Aim 2](#_Toc88701635)

[Objectives 2](#_Toc88701636)

[Survey Of Literature 3](#_Toc88701637)

[Introduction 3](#_Toc88701638)

[Mobile Game Development 3](#_Toc88701639)

[Calory Battle AR [2] 3](#_Toc88701640)

[RPG Fitness Game [4] 3](#_Toc88701641)

[AR Technologies 4](#_Toc88701642)

[ARKit [6] 4](#_Toc88701643)

[ARCore [8] 4](#_Toc88701644)

[Vuforia [9] 4](#_Toc88701645)

[AR Foundation [5] 5](#_Toc88701646)

[Technologies Conclusion 5](#_Toc88701647)

[Requirements, Outline of Specification and Design 6](#_Toc88701648)

[System Architecture 6](#_Toc88701649)

[Requirements 7](#_Toc88701650)

[Navigation 7](#_Toc88701651)

[Challenges 7](#_Toc88701652)

[Encounters 7](#_Toc88701653)

[Monster Log 8](#_Toc88701654)

[Database 8](#_Toc88701655)

[Planning and Timescales 9](#_Toc88701656)

[Semester 1 9](#_Toc88701657)

[Semester 2 10](#_Toc88701658)

[References 11](#_Toc88701659)

# Aims and Objectives of the project

## Aim

The primary aim of the project is to develop a mobile game utilizing AR technology that would allow users to interact with AR (alternative reality) objects with the help of mobile phones. This feature would be used to encourage users to exercise more by walking around towns fighting monsters.

The project would be fundamentally similar to Pokémon GO. However, this project will look thematically different. The android game will be leaning more toward Dungeon & Dragons visual RPG (Role Playing Game) game and instead of capturing Pokémon’s users would have to fight fantasy monsters such as slimes by partaking in a fun overhauled version of rock paper scissors game. The user’s goal would be to defeat all the monsters and complete their monster log.

However, to defeat all the monsters, users will have to start by fighting smaller monsters like slimes to increase their strength by gaining experience for every defeated monster. And after acquiring enough experience player character levels up and the statistics of the character are improved. Afterwards, they can challenge more difficult monsters. This process repeats until thse player can challenge the strongest monster in the game – a dragon.

## Objectives

1. Learn how to use the following technologies:
   1. C# Fundamentals for Unity programming.
   2. Unity development tool suite.
   3. Detect phone position in real-time.
   4. Use of GPS (Global Positioning System) [1] to track users’ real-time location.
2. Find free assets for the game to fit the theme.
3. Write customs scripts to interact with AR game objects.
4. Create a database to store both monster and player information.
5. Implement Mapbox to display users’ location in real-time on a map.
6. Create other interactive elements on the map apart from monsters such as dungeons where users would have to fight a wave of monsters and see how long they can survive.
7. Create a Monster log which would allow users to track their game progression.
8. Develop tests for the game to make sure everything works as intended.

# Survey Of Literature

## Introduction

My goal in surveying the research was to learn about the development process of alternative reality games, primarily for mobile phones, especially the challenges I might have to face, and learn the techniques necessary to avoid them. It is important to note that I will be excluding anything related to HMD (Head-mounted displays) since nowadays phones are powerful enough to process and place artificial objects into real-life using a camera and an inbuilt processor of a phone.

## Mobile Game Development

In this section, I will go over research papers related to mobile game development. One of the research papers focuses on the implementation of AR-based game systems while the other one focuses on the implementation of RPG elements. These two research papers help me understand the necessary principles of mobile game development.

### Calory Battle AR [2]

This is a simple game where players must go out and find calorie bombs within a time limit which would have to be defused with the help of a smartphone. The development of the first prototype, which was made without the help of the third-party engine and only used Vuforia AR [3], introduced challenges related to “loading, processing and presenting AR content”. Additionally, scholars could not realistically interact with game objects due to the lack of a physics engine. However, some of these issues were solved in their second rendition of the prototype, which used Unity 3D as a development engine. According to scholars, the graphical expression of the objects became more precise thanks to inbuilt Unity’s shaders. Furthermore, Unity allowed them to interact with game objects a lot more realistic compared to the first prototype thanks to the physics engine. It is important to note how developers emphasize real-time feedback between code and game objects provided by Unity.

From this journal article [2] it can be deduced that when creating a mobile alternative reality game use of an engine such as Unity would improve the quality of the product and in turn, would simplify the game development process.

### RPG Fitness Game [4]

World of Workout (WoW) [4] is an RPG mobile game designed to encourage app users to exercise and track their progress via the in-game avatar. This game encompasses RPG elements such as characters statistics: speed, strength, and stamina. These statistics are used to represent how strong the in-game character is and based on these stats, users can fight monsters and receive rewards. Moreover, players can exercise in real life to improve both their health and character statistics. Exercises are accurately tracked with the help of the phones internal accelerometer and GPS (Global Positioning System) [1], therefore users do not need to manually input how many push-ups or how far they have run.

However, the architecture behind the project is the part that interested me, since the mobile AR game I am designing is also RPG-inspired. Consequently, the idea of using the accelerometer to track user movements could also be used in my application to track the shaking motion of the phone in case the AR Foundation [5] package does not provide sufficiently smooth results. Furthermore, WoW also uses the database to store user data. Accordingly, this feature will also be incorporated into the game I am developing to store monster and player statistics to make it easier to track and manage both user and NPC (Non-Playable Character) data.

## AR Technologies

This section will go over existing alternative reality technologies to help me pick the best one for the project. The choice will be based on the features, complexity and how accessible it is for testing purposes. It is important to note since this project is not meant for profit, hence I will be ignoring the pricing of AR frameworks in my decision of the SDK.

### ARKit [6]

ARKit [6] is an open-source framework created by Apple and designed for the Apple family of products such as the iPhone and iPad under iOS 11 or higher. ARKit is meant for creating alternative reality games for Apple devices. The framework has various features such as: adding and interacting with 2D or 3D objects to the live view of the phone’s camera. Moreover, the framework allows developers to track smartphones position in space in real-time. ARKit can understand the horizontal and vertical flat surfaces which can be used for integrating the virtual objects. ARKit also provides brightness estimation and static 2D image recognition which allows scanning of images in real-time and displays information on these images.

With the introduction of ARKit 2 and ARKit 3, numerous new features were created and can be read more about in research papers by scholars here [7] or in the official documentation [6]. This development framework does have all the necessary features needed for the project, such as implementation and the interaction of 2D or 3D objects, via live view of the frontal camera. Moreover, it allows phone position tracking in real-time. However, it seems like Apple does not offer the possibility to test applications on emulators and an Apple device is necessary for testing.

### ARCore [8]

According to the Scholars article [7], ARCore was initially launched as Tango SDK and was an AR-dedicated framework created by Google. Tango SDK requires smartphones with depth-sensing cameras to work. Nevertheless, since it was only working on limited devices, Google relaunched Tango under a different name – ARCore. Which, unlike its predecessor, was not limited to phones with depth-sensing cameras and works with Android Nougat (7.0) and up.

ARCore [8] is an open-source framework designed for Android and iOS systems. The framework was developed by Google and is intended for alternative reality development. This framework shares a lot of similarities with ARKit, such as motion tracking, surface understanding, and brightness estimation. However, ARCore, unlike ARKit, is not limited to one operating system and does support both Android and iOS development. Therefore, ARCore provides tools to create applications for both operating systems simultaneously. Moreover, when it comes to software testing, computers can emulate the Android environment and the application can be run and tested on the computer. Hence, the android device is not necessary for the testing purposes of development.

### Vuforia [9]

Vuforia [9] is an open-source AR framework designed to work with Android and Apple operating systems which incorporates features [3] both from ARKit [6] and ARCore [8] , such as interacting with 2D and 3D virtual objects. However, there seems to be a heavy focus on industrial application for this framework since it is capable of recognizing objects by shapes such as industrial equipment, vehicles, toys, and home appliances. Vuforia can be used within its development engine or as an extension of Unity. Moreover, this framework seems to be too complex for project purposes.

### AR Foundation [5]

AR Foundation [5] is a set of MonoBehaviours (MonoBehaviour is the base class from which every Unity script derives) and APIs developed by Unity Technologies. The AR Foundation allows developers to create augmented reality projects in a multi-platform way within Unity Engine [10]. Besides, the package allows developers to use both ARCore [8] and ARKit [6] features in their development similarly to Vuforia [3], although with a focus on game development. Additionally, creating projects with the help of this package allows the creation of applications both for Android and iOS devices simultaneously. Which means it is not limited to one operating system. Because the AR Foundation is part of the Unity engine, it allows developers to have live feedback on their game development through Android emulation within the engine. This feature significantly increases the productivity of the project since developers do not need to build and run the application to test every script.

### Technologies Conclusion

Based on the researched frameworks and packages, the AR Foundation and Vuforia show to be the most fitting augmented reality packages to develop the mobile game project. Due to the fact they both can incorporate features from ARCore and ARKit. Furthermore, both packages are capable of developing applications for both Android and iOS platforms. However, since Vuforia is intended to be used in an industrial environment, unlike the AR Foundation, which was designed with game development in mind, The AR Foundation will be used in the project. Nevertheless, because, AR Foundation is part of the Unity Engine, the mobile game will have physics and possess more realistic interaction with the virtual game objects and the testing process will be simplified.

# Requirements, Outline of Specification and Design

This section will discuss the specifications of the project and explain the underlying systems of the game and the actual software needed to develop the features required. Since the project must be like Pokémon GO, the application must incorporate both AR and map features to navigate players through the streets. However, since the project must be unique, some of the core features of the project will be unique and differ from Pokémon GO. It is important to note that since the project is not meant for profit, the pricing of used software will be ignored since all the software that is used in this project fits within the free tier ranges of the proposed software.

## System Architecture

The core system is built using Unity 2019 [10] with the help of the AR Foundation package [5] which allows me to display and interact with alternative reality objects while using a smartphone. The systems as shown in Figure 1 are divided into four different levels: User interface, tools, logic, and data. The first level of the User displays the Mapbox interface for users in the form of a static map and displays the nearby monsters. Furthermore, the second level is just necessary tools for the game and the map to function. Additionally, the third level – Logic, describes how the game encounters work and interact with users. Lastly, the data level stores all information about the monsters and players in a JSON [11] format as a medium.

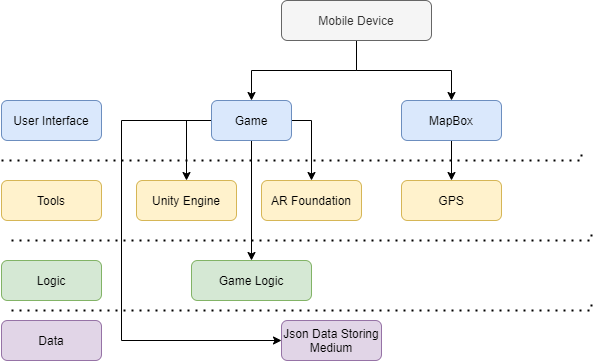


Figure 1 System architecture

## Requirements, Specification and Design Principles

### Navigation

The project will be using the phone’s GPS [1] which is a U.S.-owned utility that provides users with positioning, navigation, and timing (PNT) services. Furthermore, to encourage players to exercise, the application will also have the same core mechanics as Pokémon GO, where players must look at the map-like interface to find nearby creatures. This graphical interface and navigation system will be implemented with the help of Mapbox [12].

Mapbox [12] is a strong tool that gives developers the ability to use satellite imagery, static and print maps which could be made to be interactive and are fully customisable. This API (Application Programming Interface) will be used to help users track their current location on the map and display the nearby monsters and challenges they must walk to fight them.

### Challenges

The world map is filled with monsters and dungeons which act as points of interest for players. Every monster and dungeon will have different difficulty levels to give users a sense of a challenge. Every monster will have its unique statistics (strength, defence, agility, and luck) based on the statistics, the difficulty of every encounter will vary and create a challenging and fun experiences. However, dungeons are meant for more experienced players where they can challenge waves of monsters and try to clear the dungeon for a reward.

Both monsters and dungeons can be implemented with the help of Mapbox API [12] using the add custom markers [13] feature, which allows implementation of dungeons and monsters as custom markers on the world map where players can easily find them. The implementation of monster encounters and dungeon mechanics will be discussed further in the Encounters section.

### Encounters

Making encounters purely based on a rock-papers-scissors game would make the game very repetitive and bland since the game would be mostly based on luck. However, if game statistics and resources such as health and stamina are introduced for monsters and players, simple games of rock paper scissors can be overhauled and made more fun. Changing rock-paper-scissors to sword-shield-magic creates a more RPG-themed setting and allows the introduction of more of an RPG style combat where sword beats magic and shield beats sword (Figure 1).

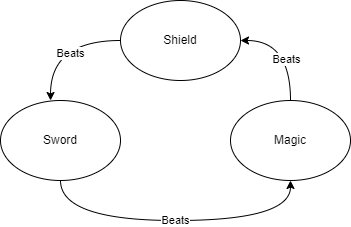


Figure 2 Visual representation of the combat system.

However, to not make it a simple rock-paper-scissors game, statistics both from the player and monster sides will be used to calculate how much damage each move does to the opponent. If the player beats the monster on a turn, they do damage based on their current statistics, such as strength, luck, and agility. But if the player does not beat it, the monster deals damage to the player. The encounter ends when either a player or the monster runs out of health. If the player wins they are rewared with experiance points that are used to level up and increase the power of the player, altough if they lose they get nothing. During a combat turn a player must perform a shaking motion akin to rock-paper-scissors with their phone and stop shaking when the desired card appears on the screen to attack the monster.This feature will be implemented to make players more involved with the game.

Since every encounter will follow the same formula of a player fighting a monster, where both sides use sword-shield-magic to attack. The combat will be handled by the Unity Engine [10] and custom writen C# Scripts [14] which would check what was picked (e.g. player picked sword and a monster picked shield) and decide the winner based on the diagram shown above (Figure 1). In a case where there is a draw, the opponent with a higher total statistical value wins.

Moreover, implementing the phone shaking mechanic to pick an action will be achieved with the help of the AR Foundation [5] device tracking feature, which allows finding the position and orientation of the phone in physical space. To identify phones positions accurately, a phone’s camera is used to distinguish interesting points called features and tracks how those points move over time [8]. Hence, this tracking can be applied to detect whether the phone is being shaken or when it is motionless.

### Monster Log

Players start the game with zero per cent completion of the monster log, and after every victorious battle against a new monster, the monster log gets updated, and the percentage gets increased up to one hundred per cent. All players should aim to complete this monster log since it acts as a progress meter to see how much of the game has been completed.

The implementation of the monster log would mostly be visual since every monster has a variable called defeated which equals false by default. The script would have to go over the list of existing monsters in the database and check whether the defeated value equals True and if it is True display the monster in the log. But if it is False, ignore it. Furthermore, the calculation for the percentage of completion is simply the number of defeated monsters/number of monsters \* 100.

### Database

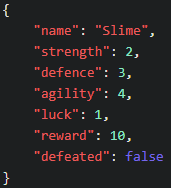
The mobile augmented reality game will have a database to store both player and monster data, since this game will be locally running on the phone there is no need for a database running on the server like in the previously discussed World of Workout [4] game where they used MySQL on a server which was intended to store data of thousands of users. However, in the case of this project, a local and lightweight data storage medium based on JSON (Figure 2) [11] would suit better since the game is not being designed with online functionality in mind.

Figure Example of JSON data storing for a Slime Monster

# Planning and Timescales

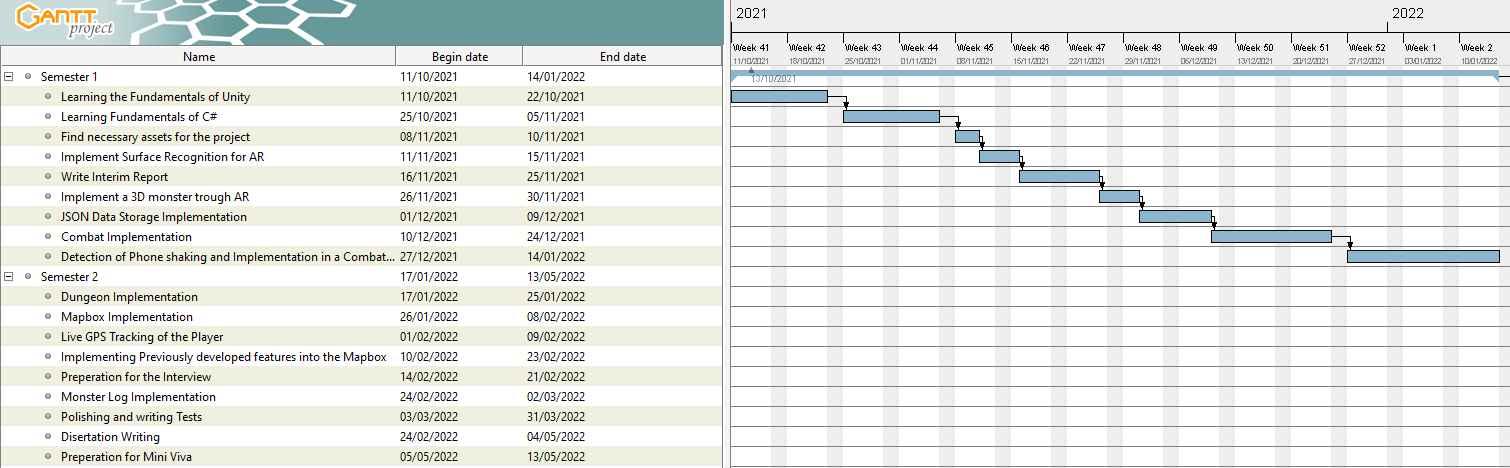


Figure 4 – Overview of the Gantt chart

## Semester 1

My primary goal for the first semester was to lay the groundwork for the second semester by acquiring the necessary assets and knowledge needed to create the game. As it can be seen in the picture above, I have dedicated around a month learning the fundamentals both of Unity [10] and C# [14]. Without spending this time, I could not have started the project, since Unity uses scripts based on C# and if I did not know the language, I could not write or read scripts necessary for feature execution such as object placement and interaction.

After Acquiring the knowledge needed for the project, I have spent a couple of days finding the necessary assets for the game development. This step not only helped me to have a better understanding of what my game should be like but also inspired me to add additional features to the game, such as dungeons to make it more challenging and entertaining. Some monster asset examples can be seen in the picture bellow – Figure 5.

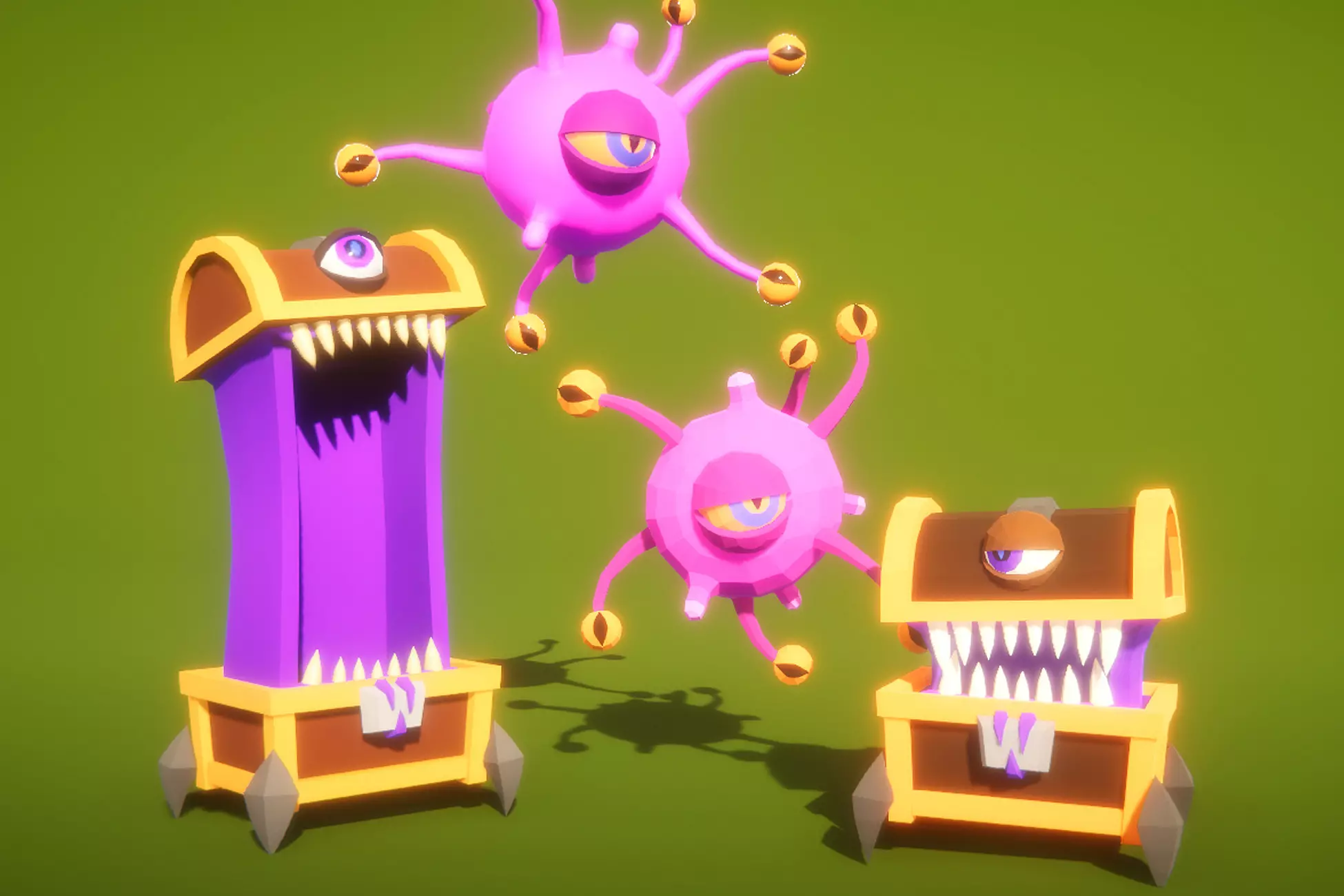


Figure 5 Example of Unity assets for the game (Mimic and a Beholder)

I have achieved everything I have planned so far. The application is currently capable of detecting surfaces with the use of the frontal camera of the phone. These surfaces will be used later to place monsters and interact with them. Since most of the project's initial stages are purely research-based, progress might seem small. However, after the submission of the interim report the actual groundwork starts for the project.

As can be seen in the figure 4 diagram, I will be implementing the acquired assets into the game using the AR Foundation [5] starting on the 26th of November. This step is crucial for the future implementation of combat where users will be able to see the monster they are fighting. However, has allocated extra days for me to fully familiarize myself with the implementation and interaction of 3D assets. Although the most challenging features to implement will have to be the combat and the phone shaking detection, that’s why these functionalities have been dedicated a month of development time.

## Semester 2

The beginning of the second semester builds on top of the first semester's groundwork. For dungeons to work, combat encounters must already be developed and fully working. I have allocated only a week to dungeon implementation because it uses the same systems that combat has but on a bigger scale. However, immediately after it, the implementation of Mapbox has been assigned a month, though it is split into three smaller parts: the interface implementation, GPS tracking, and feature merging with the Mapbox interface. In cases where a month is not enough, I can use the time designated for the polishing and testing phase. Therefore, if this step goes out of the initial time frame, I still should be good for the rest of the projects.

Furthermore, after all previously developed features are implemented and working with Mapbox. The creation of the monster log should not take longer than the estimated week since the base principles of the monster log is quite simplistic. Afterwards, the rest of the 2 months will be mostly spent on developed game system polishing and dissertation writing.

# References

|  |  |
| --- | --- |
| [1] | “GPS.gov,” [Online]. Available: https://www.gps.gov/systems/gps/. |
| [2] | S. L. Kim, H. J. Suk, J. H. Kang, J. M. Jung, T. Laine and J. Westlin, “Using Unity 3D to facilitate mobile augmented reality game development,” *2014 IEEE World Forum on Internet of Things, WF-IoT 2014,* pp. 21-26, 03 2014. |
| [3] | “Vuforia Engine Overview,” [Online]. Available: https://library.vuforia.com/getting-started/vuforia-features. [Accessed 18 11 2021]. |
| [4] | J. Bartley, J. Forsyth, P. Pendse, D. Xin, G. Brown, P. Hagseth, A. Agrawal, D. W. Goldberg and T. Hammond, “World of Workout: A Contextual Mobile RPG to Encourage Long Term Fitness,” in *Association for Computing Machinery*, 2013. |
| [5] | “About AR Foundation,” [Online]. Available: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html. [Accessed 18 11 2021]. |
| [6] | “Apple Developer Documentation,” [Online]. Available: https://developer.apple.com/documentation/arkit/. [Accessed 18 11 2021]. |
| [7] | Z. Oufqir, A. E. Abderrahmani and K. Satori, “ARKit and ARCore in serve to augmented reality,” *2020 International Conference on Intelligent Systems and Computer Vision (ISCV),* pp. 1-7, 2020. |
| [8] | “Overview of ARCore and supported development environments,” 18 11 2021. [Online]. Available: https://developers.google.com/ar/develop. |
| [9] | “Getting Started | VuforiaLibrary,” [Online]. Available: https://library.vuforia.com/. [Accessed 18 11 2021]. |
| [10] | “Documentation,” [Online]. Available: https://docs.unity.com/. [Accessed 18 11 2021]. |
| [11] | “Introducing JSON,” [Online]. Available: https://www.json.org/json-en.html. [Accessed 22 11 2021]. |
| [12] | “Mapbox,” Mapbox, [Online]. Available: https://docs.mapbox.com/help. |
| [13] | “Add markers,” [Online]. Available: https://docs.mapbox.com/unity/maps/guides/add-markers/. [Accessed 19 11 2021]. |
| [14] | “C# documentation,” [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/. [Accessed 19 11 2021]. |