COMP4801 Final Year Project

Combining Physical and Virtual Gaming Experience

Final Report

Chun Sin Ying
3035371850

Supervisors
Dr. Chim, T. W.
Mr. David Lee

Abstract

Since the gaming industry is becoming saturated, players tend to raise their requirements to the quality of gaming experience. Fusion of physical and virtual gaming experience may provide a fresh feeling to the players. In this project, a single-player Augmented Reality (AR) mobile game called Hexplore Fort is developed, where the players can control a spider-like robot - hexapod to adventure with the storyline in the game. Therefore, after developing a robot controlling application, it will then be transformed into a plugin for use in the AR gaming environment for further development of the game. The game design is finalized, and the development of game mechanism is completed. Players can explore the fort that is augmented by AR by controlling the movement of hexapod, collecting items, fighting with enemies and buying assistance provided by the spy, in order to capture the princess. Regardless of encountering some technical difficulties, the project is completed on schedule with the scope as same as stated in the proposal. Some directions of future development are suggested based on the reference to the game Tower of the Sorcerer.

Acknowledgement

I am particularly grateful for the patient guidance given by my supervisors, Dr. Chim, T. W. and Mr. David Lee. I wish to thank Dr. Chim for his valuable and constructive suggestions based on his profound experience in game design for planning the project, and all the assistances about the project assessments. Meanwhile, I wish to thank Mr. Lee for his technical support which enrich my knowledge of developing application, mechanics of hexapod used in this project and also the usage of AR technology. I would like to show my appreciation to everyone who has given assist and precious feedback to this project.

Table of Content

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
AR	Augmented Reality
ATK	Power of Attack
DEF	Power of Defense
EXP	Experience
HP	Health Point
LV	Level
QTE	Quick Time Event
RPG	Role-playing Game
SDK	Software Development Kit
SPP	Serial Port Profile
UUID	Universally Unique Identifier

1 Introduction

Physical gaming experience stands for any physical objects for entertainment, such as teddy bears, robots, etc. Meanwhile, virtual gaming experience includes digital games, AR, etc. After clarifying the definitions of physical and virtual gaming experience, the project is then narrowed down by specifying to use the combination of robot and AR game. To introduce the project, a brief background on gaming industry, AR game and robotics is given below. The objectives of the project are then provided, followed by the contributions of the project. There is an outline of this report at the last of this chapter.

1.1 Background

According to The State of Online Gaming 2019 research report from the Limelight Networks, people are spending more time on gaming. An average of 7.11 hours is spent on playing video games each week among every age group in 2019 which is higher than that of 5.96 hours in 2018 [1]. In the report of Betting on Billions, 50% of respondents, who use smartphone or tablet within 7 days during the survey conducted in December 2018, play games. As a result, game is in the top 3 most-used app type [2]. These statistics are showing that people are having an increasing demand of higher quality gaming experience, otherwise, they will easily get bored with massive similar gaming experience that have been launched to the market.

Pokémon Go is a famous AR game, which has the most daily revenue with USD$1,644,157 among iPhone mobile gaming apps in United States in September 2019 [3]. In Pokémon Go, players are required to capture the wild Pokémon that is augmented into the real-world environment (see Figure 1). However, the Pokémon is generated according to the location, and it is not specified to any object displayed in the real-world. As a result, switching off the AR mode can still be able to continue with the gameplay (see Figure 1). Since it is not necessary to switch the AR mode on for the gameplay and switching it on consumes the power of the mobile phone quickly, players tend to switch it off despite losing the interactive experience brought by AR. In this project, the robot used is one of the characters for the game. Therefore, the AR technology has a correlation to a real-world object, so as to enhance the interaction between the physical world and virtual world, and to provide a better gaming experience for players.

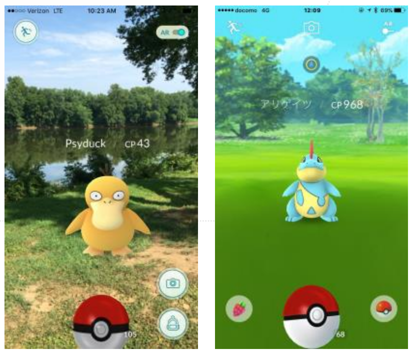


Figure 1: Screenshots of Pokémon Go; with AR (left) [4], without AR (right) [5]

On the other hand, although entertainment robots are toys with long history, they are not out of date. Instead, the market of entertainment robots, especially those are programmable so that a series of actions can be performed, is increasing because Artificial Intelligence (AI) development is growing rapidly. The market is expected to increase at a compound annual growth rate (CAGR) of 23.06% [6]. Programmable robots are the current trend, while Boston Dynamics is told to be a world leader. Boston Dynamics develops robots which can have sophisticated and smooth movements [7] (see Figure 2). However, a robot armed with excellent movements may not be interesting and useful if it is not used for any purpose. Therefore, this project merged the AR technology with the programmable robot into a mobile game.

Figure 2: Robot named Atlas developed by Boston Dynamics [8]

1.2 Objective

This project aims to provide a new gaming experience to the players as well as fulfill the increasing demand of better gaming experience by combining AR technology and robots.

Most of the recent gaming experiences are monotonous, in which they are either a physical or a virtual experience. As times goes on, the development of game industry is saturated. People get bored easily since brand-new gaming experience is difficult to be discovered for either physical or virtual entertainments. Hence, this project is mixing physical and virtual world to provide an interesting and fresh idea. The idea of combination may become new blood to the industry.

Combining the two complements the weakness each other. A robot provides a realistic vision of the movements, it can also be touched in the real-world. These are the advantages of playing a robot, however, it is not very interested to look at a robot moving around. On the other hand, AR augments computer-generated characters to the real-world environment. Although the characters are generated virtually and can only be viewed through the screen of the devices, this provides a fancy feeling to the gaming experience, and increases user interaction between the physical and virtual world. Integrating the strengths of both physical and virtual gaming experience, the result of this project is found to be providing a more attractive and interesting new gaming experience to the players.

1.3 Contributions

In this project, Hexplore Fort, a mobile game consisting AR gaming environment that can fully control the movement of the robot in physical world was developed. It is a single-player RPG, where the robot is the character for the players. The game recognizes the physical robot with the AR technology, 3D effects like shooting, burning and generating protective shield are augmented onto the robot. The mechanics of the game is inspired by Tower of the Sorcerer (see Figure 3).

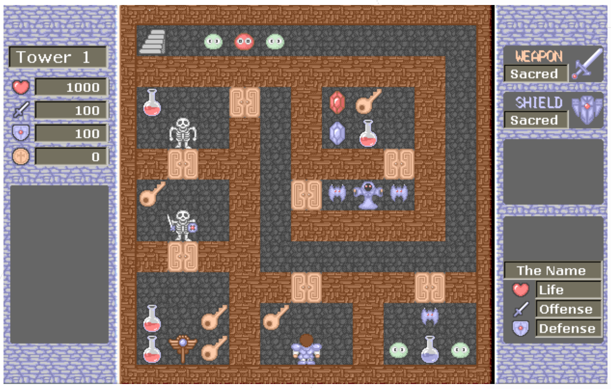


Figure 3: Screenshot of Tower of the Sorcerer [9]

It is a classic RPG developed in Japan and first issued in 1996. The feature of fixed-value RPG is reserved in the deliverable of this project, that means map, number of enemies, resources provided are not randomly generated. Since most of the value is fixed, choosing a correct order of the combats is deterministic for the players to pass through the levels, so it is also a puzzle game [9]. However, the story is changed where players are in a role of villain who breaks into the mansion to capture the princess. In the mansion, players are required to fight with some guards who try to save the princess. However, since the original combat system is of fixed value, the player's skill is often irrelevant to the outcome of the combat. Therefore, in Hexplore Fort, QTE was added in order to modernize the game's combat system and to add another challenge for the players. QTE requires players to enter a series of input manually in a fixed time. It tests the player's reaction time and precision. By utilizing ARCore, which is a development tool used for AR technology, virtual terrain with enemies, 3D effects for the robot are augmented into the virtual gaming environment. An exciting plot with a role that matched with the spider-like hexapod to be used is believed to make Hexplore Fort become more attractive.

1.4 Report Outline

The remaining of this report is organized as follows. Chapter 2 details the methodology to talk about how the project is implemented, while the project results are stated in Chapter 3. Chapter 4 then describes the limitations and difficulties encountered during the project. In Chapter 5, any possible further development can be completed in the future are stated. A conclusion to sum up the project are provided in Chapter 6.

2 Methodology

The project is firstly divided into two parts, mobile application to control the robot and game with AR gaming environment. After ensuring each part of them is working properly, merging of two parts is carried out. To be exact, the mobile application is modified into a plugin that can be used by Unity, the game engine that used to develop for the AR gaming environment. Some specifications on hardware and software are stated below, followed by the details of each part of the project.

2.1 Equipment and Set Up

The robot used is a hexapod, which is a spider-like robot with six legs (see Figure 4). It is a 3D-printed Arduino-based robotics and equipped with Bluetooth serial board, so that it can connect to other device through a wireless personal area network. Bluetooth Classic is used with SPP because it has a higher compatibility to support more devices while compares to other Bluetooth technology. In order to connect with the robot, a mobile phone which can support Bluetooth connection and location service is needed. More than that, supporting AR technology by ARCore, which is a plugin developed and maintained by Google, with Android system is also required for the phone. After equipping the required hardware, a plenty of space with ground is needed for testing with the deliverable.

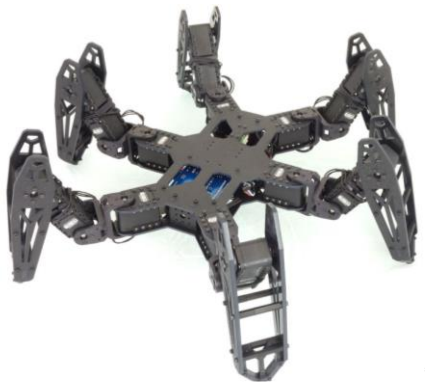


Figure 4: A hexapod from Trossen Robotics [10]

2.2 Software Specifications

Since Android is used due to the simplicity of public API for Bluetooth Classic in Android compare to Core Bluetooth for iOS, Android Studio with Java language is used to develop the mobile application to control the robot. Meanwhile, Unity with C# language is used for building the gaming environment. Another way of implementing this project is using Android Studio for both controlling the robot and gaming environment. However, Android Studio is not specified for game development. Compared to Unity which is a famous game engine that used to develop game, Unity has more features and a better user interface to ease the game development. For example, Unity supports an easier implementation of animations, and it can control the locations of the game objects in the virtual 3D world easily. ARCore SDK is integrated in the Unity game engine, therefore, ARCore is used to support the AR technology in the game.

2.3 Robot Controlling

Hexapod used in this project can have 60 kinds of movements (see Figure 5). There are 3 categories of modes, that is walking (W), dancing (D) and fighting (F).

	A	B	C	D
W (walk)	Low step	High step	Small step	Scamper
D (dance)	Freestyle	Ballet	Waves	Hands
F (fight)	Front legs	Front legs, unison	Swivel	Lean

Figure 5: Modes available for hexapod movements [11]

4 modes are in each category, while 5 directional buttons indicating the direction of movements. A user layout that is similar to the original Vorpal gamepad (see Figure 6) of the robot is used to implement the complicated controlling method in the robot controlling application.

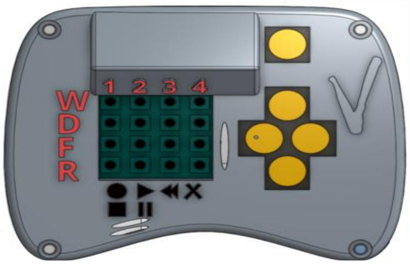


Figure 6: Gamepad for hexapod [11]

Android Studio with Java language is used to develop the following mechanism for the application. The application first requests for the Bluetooth permission and enables the Bluetooth. The application then scans the nearby Bluetooth devices, and establishes a connection between the Bluetooth device that the user chose and the phone. After the connection is established, the application simulates as the gamepad to send out formatted Bluetooth commands to hexapod communication packets for controlling the robot movements. Hexapod will perform corresponding actions after receiving these commands.

2.4 AR Gaming Environment

ARCore supported in Unity is used. It can augment images, animation and 3D object that is responsive to the reference images in real-world environment provided by the developer by using image recognition. As long as the players move their hands and keep the object under the camera, ARCore can track moving images on the object. With this technology, effects like shooting, burning are augmented to the robot for better gaming experience (see Figure 7).

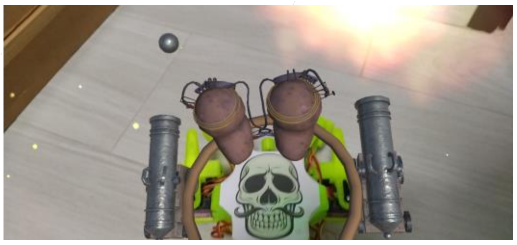


Figure 7: Augmented 3D effects to hexapod

Apart from augmenting effects to the robot, the virtual terrain of the mansion with items and enemies that the players have to fight with are augmented by recognizing the flat surface of the floors by the plane detection provided by ARCore. The virtual terrain is generated based on the anchor point which describes a fixed location and orientation in the real world recognized by ARCore, this makes the terrain fixed on the floor (see Figure 8). Therefore, the physical robot can act like walking in the virtual terrain

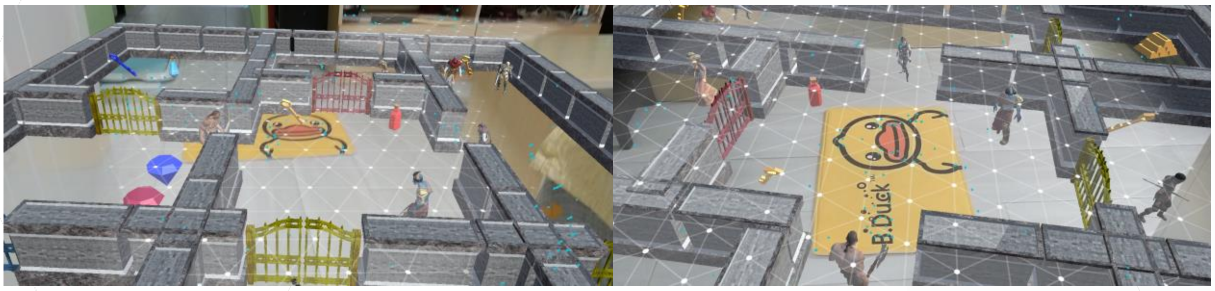


Figure 8: Augmented terrain in different view angle

Using the above two recognition technology of ARCore, the game design is stated as follows. Players have to control the robot to walk through the augmented map for the mansion, picking up items such as keys to unlock the doors on the map, and gem to increase the power of attack and defense for fighting with the augmented computer-generated characters. In practical, some animations such as burning and 3D assets such as walls, characters like knights are designed and rendered into the virtual gaming environment by Unity after using ARCore to recognize the robot and the flat surface.

2.5 Combining Robot Controlling and AR Game

Since Unity supports importing Java source files as plugin for Android, only minor adjustments to the application for robot controlling is needed in order to convert it into a plugin that can be used by Unity. However, it is impossible to have all 60 kinds of movement supported in the game due to the complexity of controlling method for a game. Some of the movements are selected and a simple interface is built in Unity for apply these movements. Each button and joystick in the layout will correspond to relative Bluetooth commands built in plugin. Therefore, the mechanics are as follows. After pressing the button or controlling the joystick in AR game built by Unity, Unity would receive an event to call the relative function from the plugin for Android. Bluetooth commands would be sent out by Android plugin and received by the robot to perform corresponding actions.

3 Discussion of Results

In this chapter, the results of the project are discussed in detail. The following includes the descriptions of the robot controlling application, conversion of it into the plugin of Unity and also the game Hexplore Fort with some results of player game testing carried out.

3.1 Robot Controlling Application

The user interface of the controller is solely for demonstration purpose and is replaced by corresponding controls of interface built in the AR game by Unity. Since the controller requires the permission to access the Bluetooth and location services of the phone, the permission requirements are added into the **AndroidManifest.xml** which is a file included in every Android application project for describing essential information about the application to Android build tools. The following permissions are required: **BLUETOOTH, BLUETOOTH_ADMIN, ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION**.

After granting all permissions, the controller requests for the Bluetooth permission and enables the Bluetooth to start scanning other Bluetooth devices (see Figure 9). The application scans the nearby Bluetooth device with its name and address fetched for display (see Figure 10), and the user should click to choose a device for establishing the connection. The implementation of the scanning progress took reference to the support provided by supervisors [12]. Since it is predicted that the controller should only be used to connect the hexapod. Other devices which are not using SPP are set to have a result of connection failed (see Figure 11), it is done by specifying the UUID of SPP for the Bluetooth connection socket.

```
bluetoothDevice.createInsecureRfcommSocketToServiceRecord(SPP_UUID)
```

Every object in Bluetooth has its UUID, the object could be a service, a characteristic etc. While **SPP_UUID** in the code is the SPP service's standard UUID.

```
SerialPortServiceClass_UUID = '{00001101-0000-1000-8000-00805F9B34FB}'
```

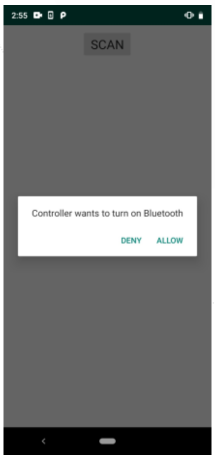
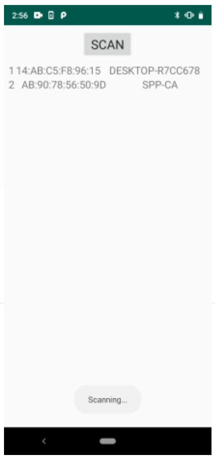


Figure 9: Controller asking to enable Bluetooth



```
1 14:AB:C5:F8:96:15   DESKTOP-R7CC678
2  AB:90:78:56:50:9D          SPP-CA
```

Figure 10: Controller scanning nearby devices

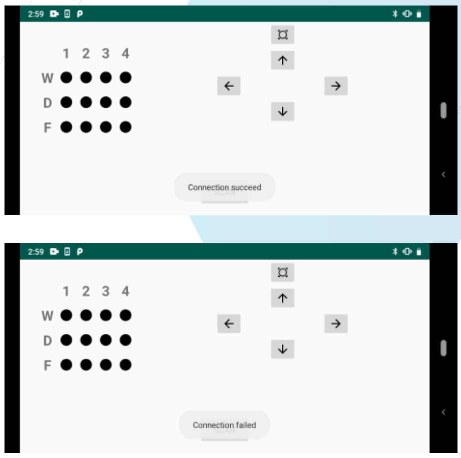


Figure 11: Connection result with device; with SPP (up), without SPP (down)

After the connection is successfully established, the controller should simulate the gamepad for having complicated control of the robot. The functions of every button are explained (see Figure 12). The initial approach of the Bluetooth communication was sending the Bluetooth commands once the button is pressed, but this may cause a continuous sending of commands. The hexapod may not have time and computation power to make response to any of the commands since it keeps receiving commands. Therefore, a regular time interval should be set to ensure only one command would be sent per time interval, even though some commands would be ignored if more than one buttons are pressed within the time interval. After trial and error, 0.2 second is found to be the best time interval to maximize the time given to hexapod for response and minimize the number of commands missed at the same time. Therefore, the application is set to send out formatted Bluetooth commands in a regular time interval for 0.2 second and hexapod will perform corresponding actions after receiving these commands.

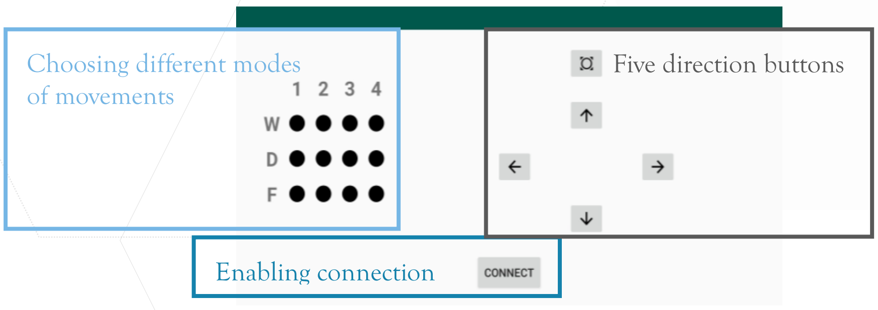


Figure 12: User interface of controller to control hexapod

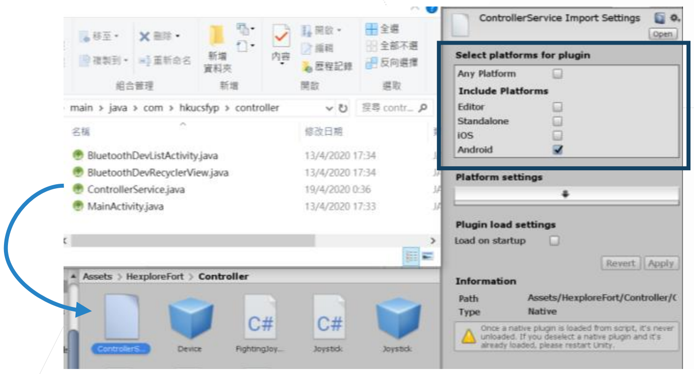
3.2 Conversion of Robot Controlling Application

In order to make the controller source code executable from Unity, few procedures are carried out, including overriding the Android manifest and requesting permissions when the game starts, importing and calling of the Java source code as plugin, last but not least, modifying the controller source code for executing some callback functions from Unity.

Although Unity generates a correct manifest for the game, importing another manifest file to certain location can have direct control over its contents and overrides the default. In this situation, Unity merges the created manifest into the main manifest, and ensures that the resulting manifest's configuration is correct. Therefore, a **AndroidManifest.xml** file is created with the same permissions stated as the manifest file in the robot controlling application. Besides stating the permission requirements in manifest file, Unity C# code for requesting the corresponding permission is created using the **Android.Permission** API to have better control of the permissions authorization. Hexplore Fort checks whether the permissions are granted and raise the corresponding request if any of the permissions is not granted by the following code when the application starts.

```
if (!Permission.HasUserAuthorizedPermission(Permission.CoarseLocation))
    Permission.RequestUserPermission(Permission.CoarseLocation)
```

Second, the functions in controller that are used in the game is prebuilt into the same class, **ControllerService** class, due to the simplicity of importing a single Java source file into Unity instead of an Android library. Therefore, the controller service class is directly imported into Unity (see Figure 13).


main › java › com › hkucsfyp › controller

名稱	修改日期
BluetoothDevListActivity.java	13/4/2020 17:34
BluetoothDevRecyclerView.java	13/4/2020 17:34
ControllerService.java	19/4/2020 0:36
MainActivity.java	13/4/2020 17:33

Assets › HexploreFort › Controller

ControllerService Import Settings

Select platforms for plugin
Any Platform ☐
Include Platforms
Editor ☐
Standalone ☐
iOS ☐
Android ✓

Platform settings

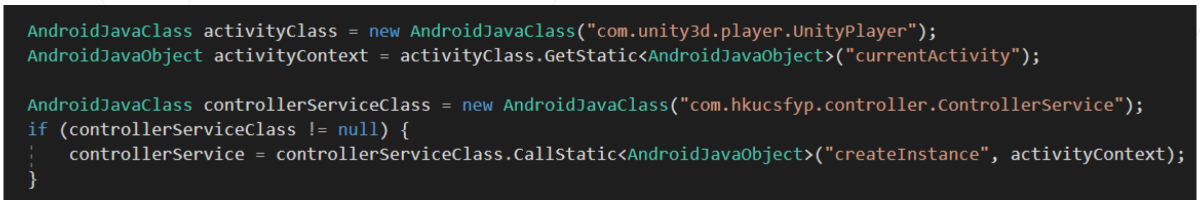
Plugin load settings
Load on startup ☐

Information
Path Assets/HexploreFort/Controller/C
Type Native

Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin and it's already loaded, please restart Unity.

Figure 13: Direct import of ControllerService.java

An instance of the **ControllerService** class is created when the application starts in the Unity C# script by the code shown in figure (see Figure 14).


```
AndroidJavaClass activityClass = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
AndroidJavaObject activityContext = activityClass.GetStatic<AndroidJavaObject>("currentActivity");

AndroidJavaClass controllerServiceClass = new AndroidJavaClass("com.hkucsfyp.controller.ControllerService");
if (controllerServiceClass != null) {
    controllerService = controllerServiceClass.CallStatic<AndroidJavaObject>("createInstance", activityContext);
}
```

Figure 14: Scripts for creating ControllerService class instance

Such that the instance is utilized to call the member functions in the plugin class by the following statement throughout the gameplay:

- bool start = controllerService.Call<bool>("startDiscovery"); //with returned value
- controllerService.Call("connectHexapod", address); //with parameters
- controllerService.Call("stopDiscovery"); //without returned value and parameters

At last, some code is modified, and the following code is added for **ControllerService** class for executing the callback functions in Unity C# script. It will find the specified GameObject in Unity scene by name and execute the specified function by name found in any component belongs that GameObject with the string parameter passed.

- UnityPlayer.UnitySendMessage(GameObjectName, FunctionName, StringParam)

Such callback functions are needed when any scan result and connection result is received, and some actions are needed to be carried out in Unity based on the result.

3.3 Game Design

As mentioned in the introduction chapter, Hexplore Fort took reference to a game called Tower of the Sorcerer. Besides of keeping the characteristic of fixed-value RPG, a similar style to the referenced game of game assets [13][14] are applied (see Figure 15). Since this game is a very classic and famous, many contributors cooperate to develop an open-sourced platform [15] for anyone to create a game which is similar to Tower of the Sorcerer, whereas the game assets are also provided [16]. Although the 2D graphics for the characters and enemies (see Figure 16) are not suitable for this project, the audio clips like background music and sound effects are included in Hexplore Fort.

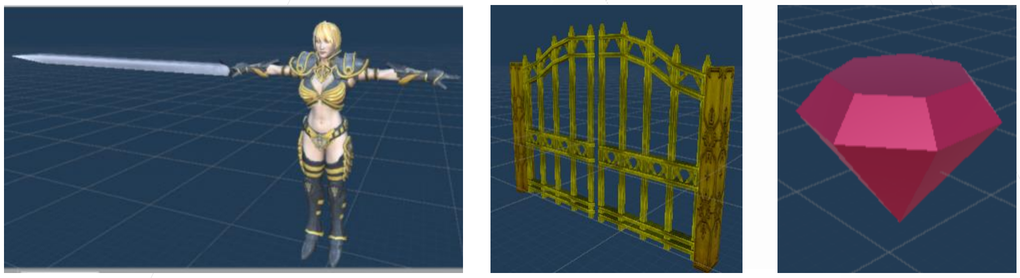


Figure 15: Style of game assets

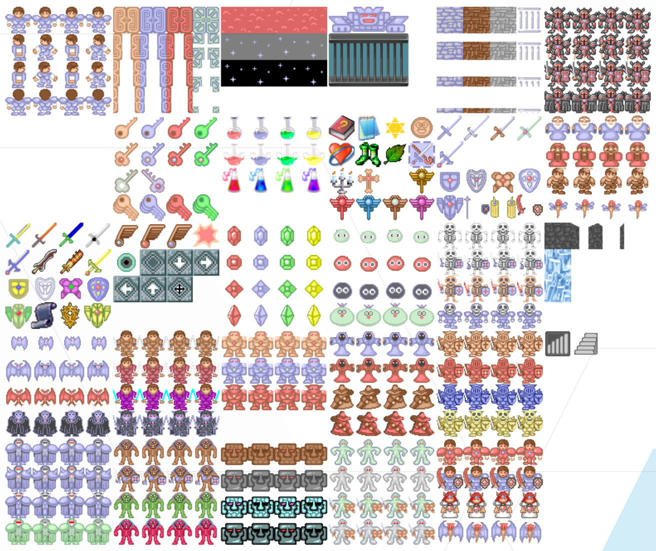


Figure 16: 2D graphics of Tower of the Sorcerer [16]

Other than reserving features from the inspiration, some changes are made to provide more excitements and fresh feeling. Hexplore Fort reversed the storyline into where players are in a role of villain who breaks into the mansion to capture the princess. QTE was added in order to modernize the fighting system and to add another challenge for the players. Since Hexplore Fort is a demonstration used deliverable for this project, the level system in Tower of the Sorcerer is eliminated in this project and Hexplore Fort is having only one fixed map, while the level system can be left as future development. Details of the storyline, game mechanics and fighting system will be discussed in later sections.

3.4 Game Flow

The flow of Hexplore Fort is described in below figure (see Figure 17).

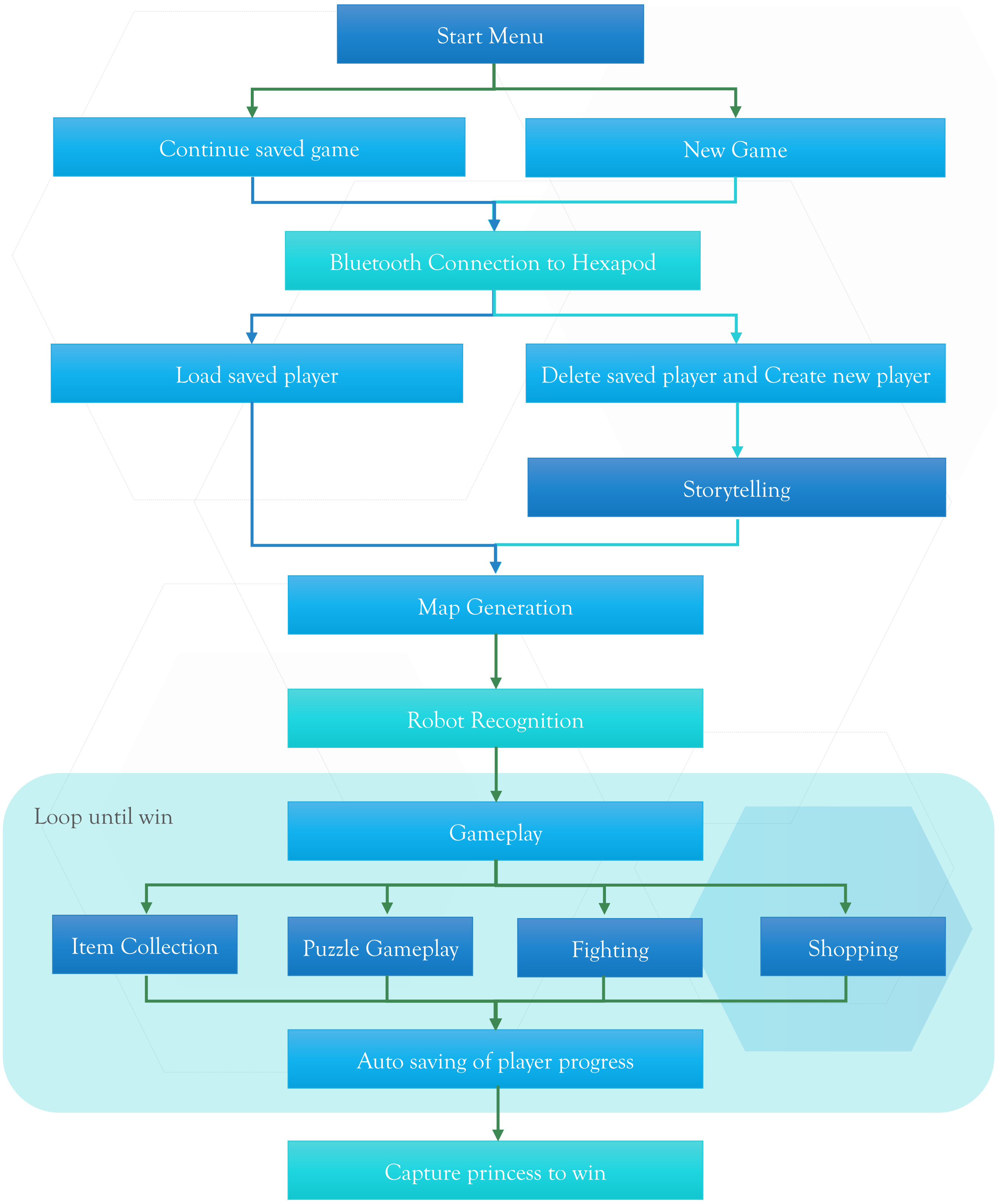


Figure 17: Game flow of Hexplore Fort

3.5 Start Menu

The start menu is composed of 4 buttons namely “Continue”, “New Game”, “Setting” and “Exit” and a background photo that used Photoshop to paste the hexapod onto a photo of ruin [13] (see Figure 18).

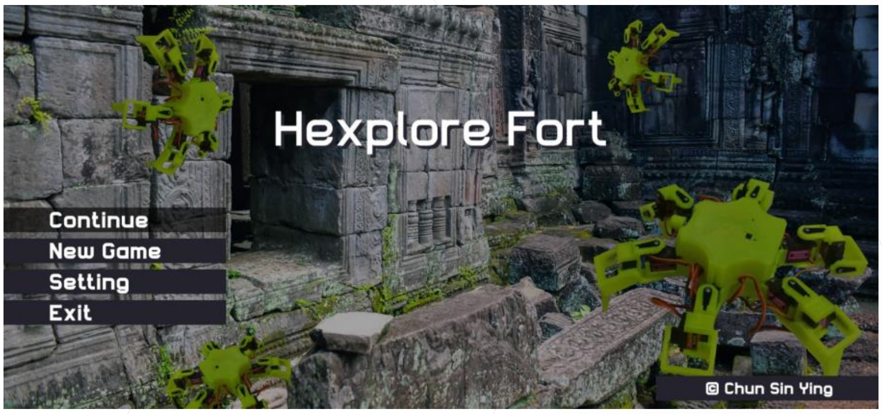


Figure 18: Start menu

When either “Continue” or “New Game” button is clicked, they will both enter the procedure of connecting to hexapod with Bluetooth, the details of this procedure will be discussed in the next chapter. However, in the case of “New Game”, after the connection succeeded, the game will delete any saved record and create a brand-new record in the device. Moreover, it will begin with the storytelling part and proceed to the main gameplay afterwards. On the other hand, in the case of “Continue”, the saved player record will be directly loaded, and players could proceed to the main gameplay right away. The details of the local saving system Other than that, the “Setting” button allows players to customize the volume of audio played in game, and the “Exit” button allows players to quit the game.

3.6 Bluetooth Connection to Hexapod

The pop-up window shown below (see Figure 19) is used to handle the Bluetooth connection to hexapod.

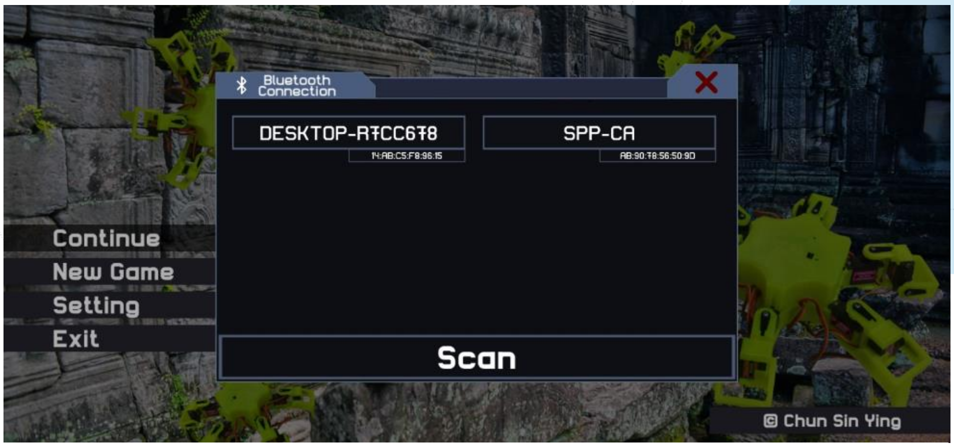
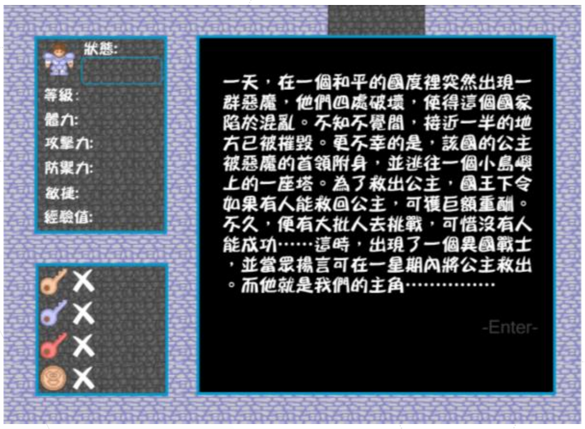


Figure 19: Bluetooth connection window

The “Scan” button is having the same functionality as that in the robot controlling application introduced in previous chapter (see Figure 9), which is enabling Bluetooth if it is not enabled, and start the discovery of nearby Bluetooth device. The function is executed by the **ControllerService** class, so a callback function in Unity is needed when there is new scan result received. The Bluetooth address and name of the device scanned is passed to Unity for displaying like the above figure (see Figure 19). Players should click on the hexapod device, which named “SPP-CA” in the example, for initiating the establishment of connection. As mentioned above, other devices without SPP will not be succeed for the connection, the next procedure will be proceeded only when the connection succeed.

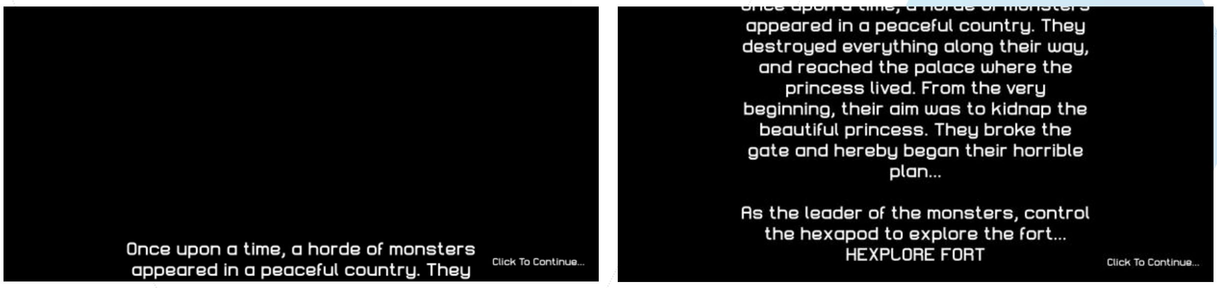
3.7 Storytelling

Hexplore Fort has a simple plot which reversed from the plot of Tower of the Sorcerer, therefore, a narrative storytelling at the beginning and the end of the game is used instead of an interactive way. The old-fashioned storytelling style in Tower of the Sorcerer (see Figure 20) is referenced to build the storytelling scene in Hexplore Fort with an animation of scrolling (see Figure 21).



一天，在一個和平的國度裡突然出現一群惡魔，他們四處破壞，使得這個國家陷於混亂。不知不覺間，接近一半的地方已被摧毀。更不幸的是，該國的公主被惡魔的首領附身，並逃往一個小島嶼上的一座塔。為了救出公主，國王下令如果有人能救回公主，可獲巨額重酬。不久，便有大批人去挑戰，可惜沒有人能成功……這時，出現了一個異國戰士，並當眾揚言可在一星期內將公主救出。而他就是我們的主角……………

Figure 20: Storytelling in Tower of the Sorcerer



Once upon a time, a horde of monsters
appeared in a peaceful country. They

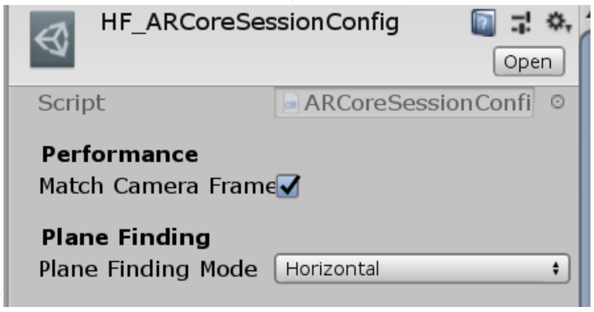
Once upon a time, a horde of monsters
appeared in a peaceful country. They
destroyed everything along their way,
and reached the palace where the
princess lived. From the very
beginning, their aim was to kidnap the
beautiful princess. They broke the
gate and hereby began their horrible
plan...

As the leader of the monsters, control
the hexapod to explore the fort...
HEXPLORE FORT

Figure 21: Storytelling in Hexplore Fort with scrolling animation

3.8 Map Generation

To start the game, players have to first find a plenty of space of flat surface using the back camera of the phone. Plane detection of ARCore is enabled in this procedure. To ensure that the map will be generated on the ground, the configuration of ARCore session is set to detect horizontal plane only (see Figure 22).


HF_ARCoreSessionConfig

Script ARCoreSessionConfi

Performance
Match Camera Frame ✓

Plane Finding
Plane Finding Mode Horizontal

Figure 22: ARCore session configuration

ARCore is looking for clusters of feature points which appear to lie on common horizontal flat surface when using plane detection, this feature is called environmental understanding performed by ARCore. ARCore also determine the boundary of each plane, so that this project can make use of the information to place the virtual terrain resting on flat surfaces. Since ARCore uses feature points to detect planes, having a surface with more texture will have a better detection result, otherwise it may not be detected properly [17] (see Figure 23). In Hexplore Fort, the feature points detected is visualized in blue color, and the detected plane is visualized by white triangles (see Figure 24).

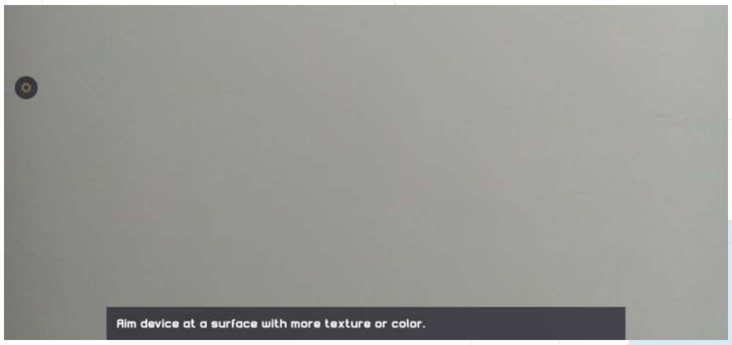


Figure 23: Plane detection to flat surface without texture

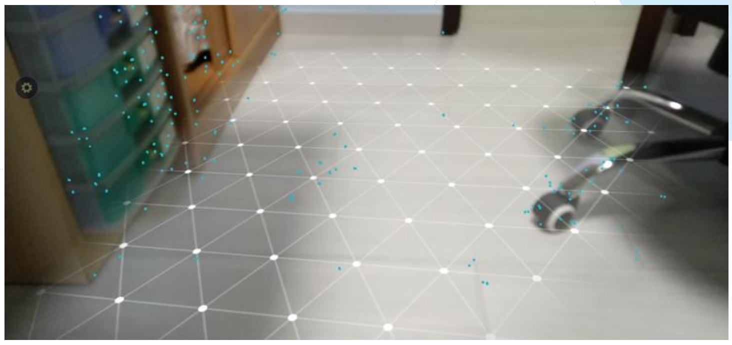
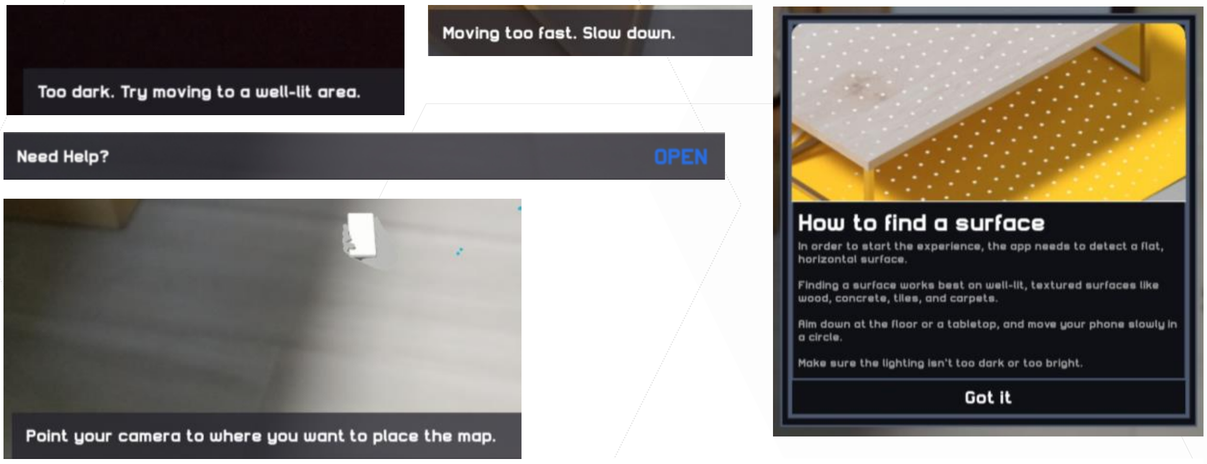


Figure 24: Visualization of feature points and detected plane

The bottom black bar is showing the guidelines for players about the situation of plane detection (see Figure 25). Players should click on the screen for where they want to place the map (see Figure 26), so as to create an anchor point at where the players point to ensure that ARCore tracks the position over time, the map will seems to be fixed on the floor as a result.


Too dark. Try moving to a well-lit area.

Moving too fast. Slow down.

Need Help? OPEN

Point your camera to where you want to place the map.

How to find a surface

In order to start the experience, the app needs to detect a flat, horizontal surface.

Finding a surface works best on well-lit, textured surfaces like wood, concrete, tiles, and carpets.

Aim down at the floor or a tabletop, and move your phone slowly in a circle.

Make sure the lighting isn't too dark or too bright.

Got it

Figure 25: Guidelines for plane detection

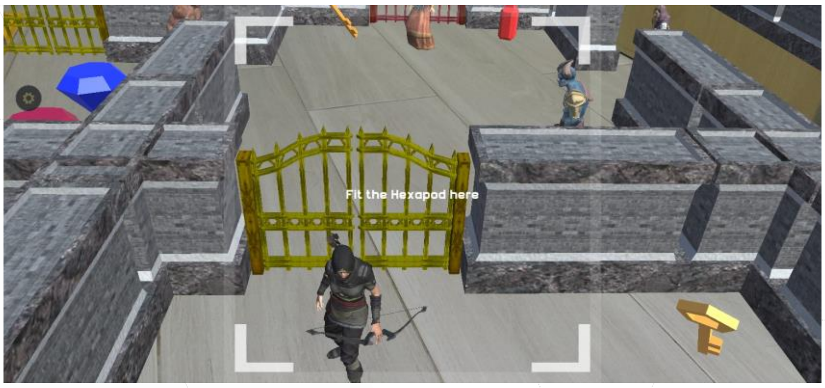
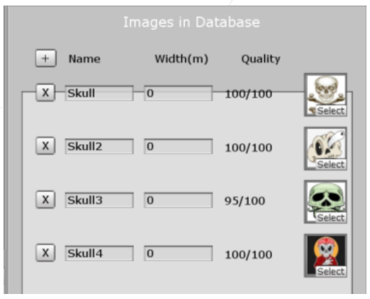


Figure 26: Map generated on the ground

3.9 Robot Recognition

After generating the map, players should recognize the hexapod on the starting point of the map. By using the image recognition in ARCore, the referenced images to be recognized are compiled offline by adding into the image database (see Figure 27).


Images in Database

	Name	Width(m)	Quality
X	Skull	0	100/100
X	Skull2	0	100/100
X	Skull3	0	95/100
X	Skull4	0	100/100

Figure 27: Image database for image recognition

The images have several requirements. It is recommended using the image with a score of at least 75. To achieve higher score, the referenced images selected for this project are at least 300 x 300 pixels, with points of high contrast, with packed features and avoided repetitive features [18]. It is stuck onto the body of hexapod and the eyes of the hexapod will be augmented onto the robot if the game recognized the robot successfully (see Figure 28).

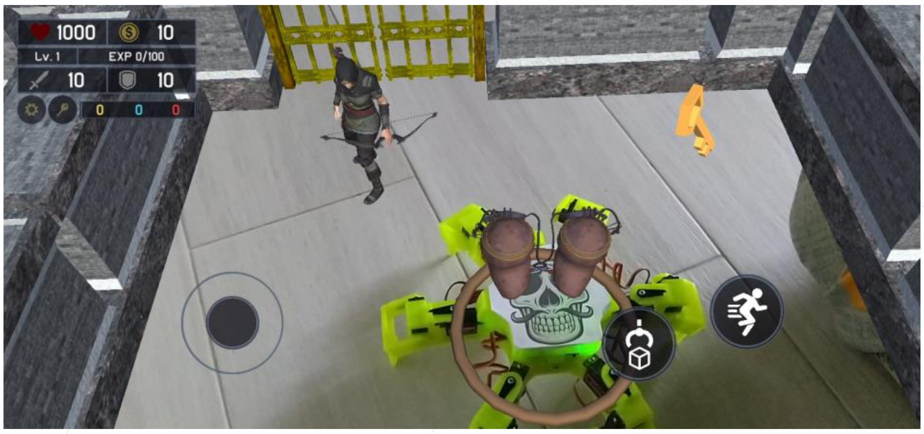


Figure 28: Hexapod recognized on the map

3.10 Player Setting

There are some statistics for the player, namely the HP, amount of money, LV, EXP, ATK, DEF, number of yellow, blue and red keys owned. These are all displayed at the top left corner of the game, and the below figure is showing the default values of these statistics (see Figure 29). The values are updated during the process of gameplay, such as after picking up items and when fighting with enemies.

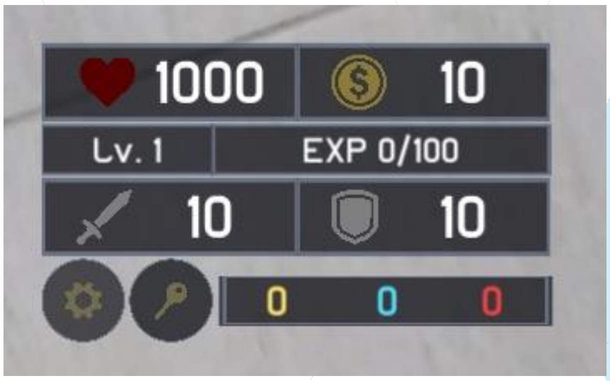


Figure 29: Player statistics shown in game

3.11 Game Mechanism

In Hexplore Fort, players are required to walk around the virtual terrain and pick up items to solve the puzzle gameplay or enhance the ability of the player character. Having a stronger ability can combat the enemies more easily. Winning from the battle can gain money which can be used to shop for increasing the ability of the player character or buying keys to make the puzzle gameplay easier. The mechanism repeats until the players win or dead. Each part of the mechanism will be discussed in following chapters.

3.11.1 Movement Control of Hexapod

There is a joystick and 2 buttons for players to perform movement control at the bottom of the game interface (see Figure 30). The up and down of the joystick is used to control the moving direction for forward and backward, while left and right is used to control the rotation of the hexapod. The walking movement is default having “Low step” mode (see Figure 5). However, if the button with running icon is pressed when controlling the joystick, the walking mode will then become “Scamper” to allow players walk faster. Meanwhile, the button with pick up icon is used for item collection.

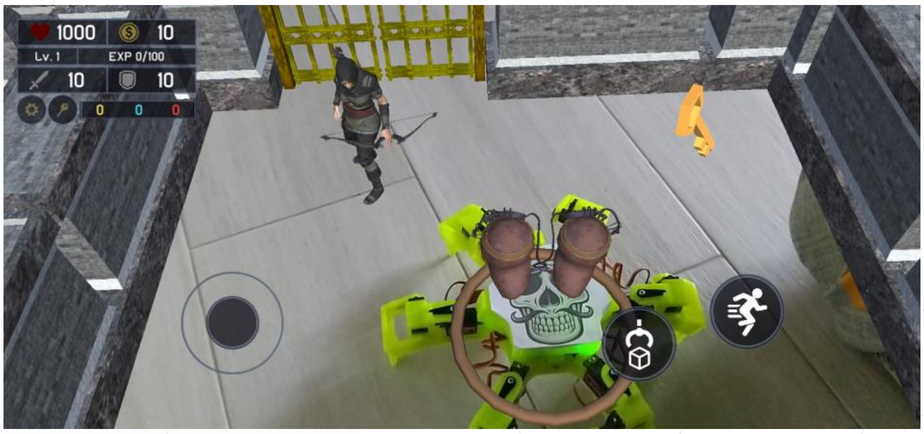


Figure 30: Game interface for movement control

A function is built to ease the modification of formatted Bluetooth command to be sent out for controlling the robot. When there is any change of movement needed to be carried out, the function **SetMovementMode** in Unity will be called with the parameters of mode letter, mode number and the direction pad letter. The **setMovement** function in **ControllerService** class is then executed with the same parameters passed to modify the local variable which indicates the movement command to be sent. The function **formatMessage** in **ControllerService** class is called in the regular time interval of 0.2 second in the use of the modified local variable. While the Bluetooth command to be sent our is formatted as stated below, while giving an example of walking forward in “Low step” mode for the variable in blanket.

- **byte representation of ‘V’ + ‘1’ + 8 + modeLetter(‘W’) + modeNumber(‘1’) + dPadLetter(‘f’) + ‘B’ + 0 + 0 + 0 + 0 + checksum(calculated sum mod 256)**

3.11.2 Item Collection

There are many different kinds of item with corresponding usage (see Table 1), players should pick up the items in order to solve the puzzle gameplay or upgrade the ability of the player character for fighting against the enemies All these resources provided are fixed and not randomly generated due to the characteristic of fixed value RPG mentioned above.

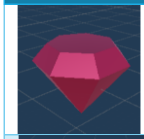
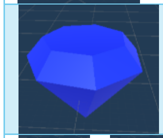
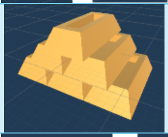
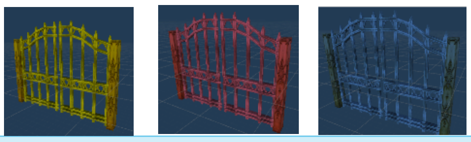
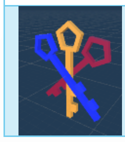
Item	Usage
 Red gem	ATK +3
 Blue gem	DEF +3
Red / Blue potion	HP +200 / +500
 Gold	Money +200
Yellow / Red / Blue key	Pick up to open corresponding colored door 
 Key box	Yellow, red and blue key owned +1

Table 1: List of items and usages

When the players control the hexapod to arrive at the location of where the item is placed, the player can click the button with pick up icon to pick up the item. The action when hexapod is in the mode of "Waves" with the directional letter "w" is selected as the action of picking up items (see Figure 31).

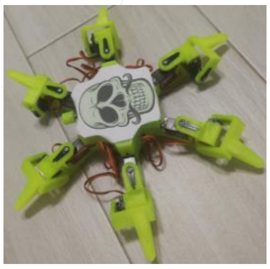


Figure 31: Pick up action of hexapod

Hexapod will perform the pick-up action for 2 seconds and automatically return to the standby walking mode with the use of **StartCoroutine** and **IEnumerator**.

```
StartCoroutin(PickUp())
IEnumerator PickUp() {
        ... // Perform pick up action
        yield return new WaitForSeconds(2f);
        ... // Perform walking standby action
}
```

3.11.3 Puzzle Gameplay

Since there are several keys placed on the map used for opening the locked door, if the players choose the wrong route and open the door not expected as the optimal solution for the puzzle, it may take longer time for the player to complete the game for having a more complexed route to capture the princess. Or even more risky as there may be stronger enemies along the more complexed route, the player may easily lose when facing the strong enemies. For example, the map is arranged as below (see Figure 32). There are two routes to capture the princess, one with shorter route and fewer enemies to be encountered along the route (see white line in figure), while the another one with longer route and more enemies to be encountered (see yellow line in figure).

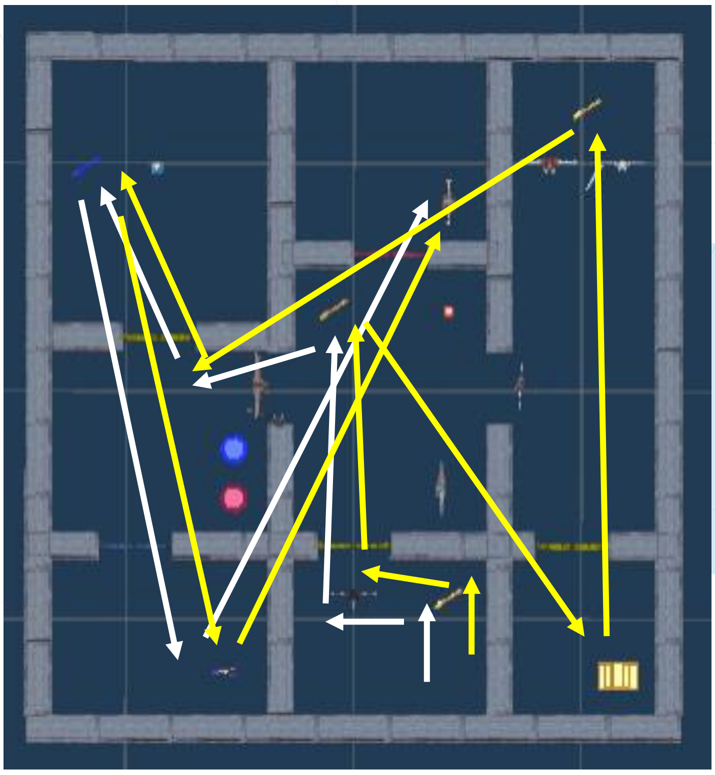


Figure 32: Floor plan of the map

3.11.4 Fighting System

There are different kinds of enemy with corresponding statistics (see Table 2), the players will not know about which enemy is having a stronger statistic before fighting with the enemy. However, the players are not allowed to escape or quit in mid-way of the fighting, so the players either beat the enemies or dead once they start the battle.

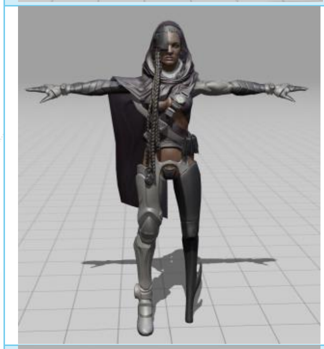
Model	Statistics
	LV: 1 HP: 70 ATK: 15 DEF: 2 Money gain after win: 2 EXP gain after win: 2
	LV: 2 HP: 150 ATK: 12 DEF: 9 Money gain after win: 3 EXP gain after win: 3
	LV: 3 HP: 200 ATK: 20 DEF: 5 Money gain after win: 4 EXP gain after win: 4
	LV: 4 HP: 250 ATK: 25 DEF: 10 Money gain after win: 5 EXP gain after win: 5
	LV: 5 HP: 300 ATK: 40 DEF: 10 Money gain after win: 6 EXP gain after win: 6

Table 2: List of enemies and their statistics

The fighting automatically starts once the players collide with the enemy. The top right corner displayed the statistic of enemy fighting with. The bottom of the game interface will change from the 2 buttons for movement control into 3 buttons for players to perform QTE. There is a line of input required for the QTE is generated randomly and displayed at the top of the interface (see Figure 33). QTE requires players to enter a series of input manually in a fixed time of 10 seconds. It tests the player's reaction time and precision and rewards the players with bonus damage to the enemies if they completed the QTE. The number of inputs required determined based on the LV of the enemies.

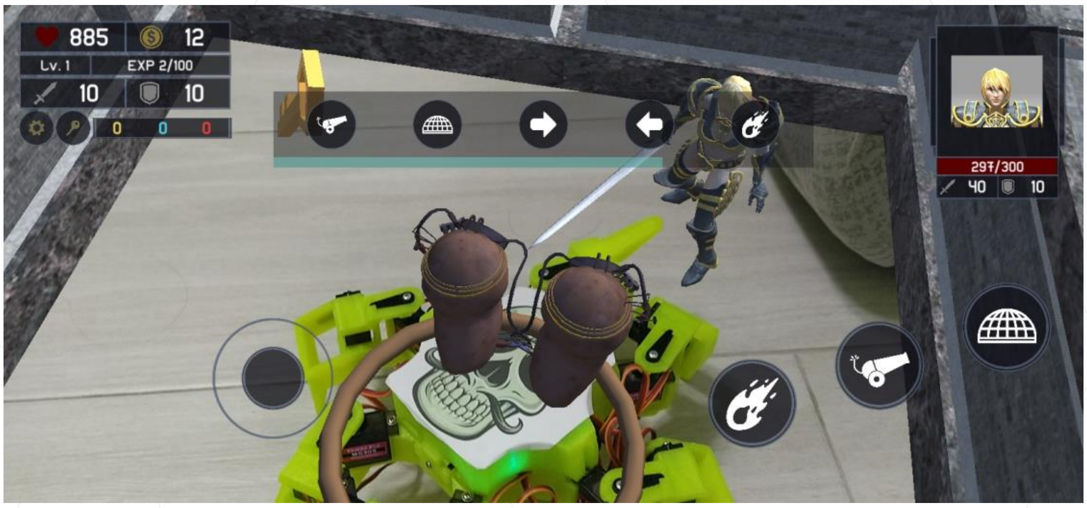


Figure 33: Game interface for fighting

The game will perform the last operation of QTE. There are 3 different operations can be performed with corresponding 3D effects augmented onto the hexapod by AR technology and movement carried by the hexapod automatically, including shooting by cannon with the movement in "Hands" mode with "f" directional button (see Figure 34), spurting fire with the movement in "Front legs" mode with "w" directional button (see Figure 35) and generating protective shield with the movement in "Swivel" mode with "b" directional button (see Figure 36).

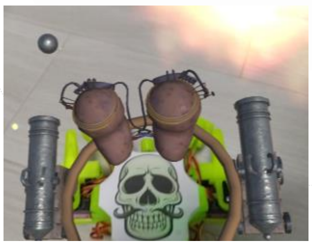


Figure 34: Shooting by cannon

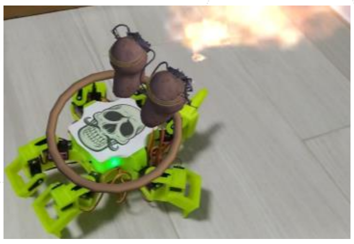


Figure 35: Spurting fire

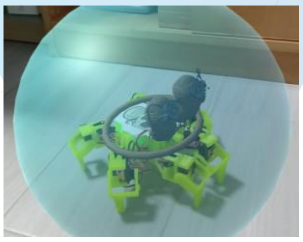


Figure 36: Protective shield

When the fighting starts, the player and the enemy will both perform normal attack with the movement in “Freestyle” mode with “w” directional button with its ATK value in regular time interval of 1 second. However, when aggressive operations are performed, such as shooting and spurting fire, ATK of players will be doubled. Moreover, when protective shield is active, enemy will cause 0 damage to the player, but if players fail to complete the QTE for activating protective shield, ATK of the enemy will be doubled. All bonus ATK added and effects of QTE operation to both characters will be reset and disappeared after 3 seconds. The formula to calculate the damages to each character is stated as follows.

- damage = Mathf.max(ATK_of_Opponent – DEF_of_Self, 1)

After winning the enemy, a random celebration actions selected will be performed automatically by the hexapod. The celebration actions include “Waves” mode with the directional letter “l” or “Freestyle” mode with the directional letter “l”, “f” or “r” (see Figure 37).

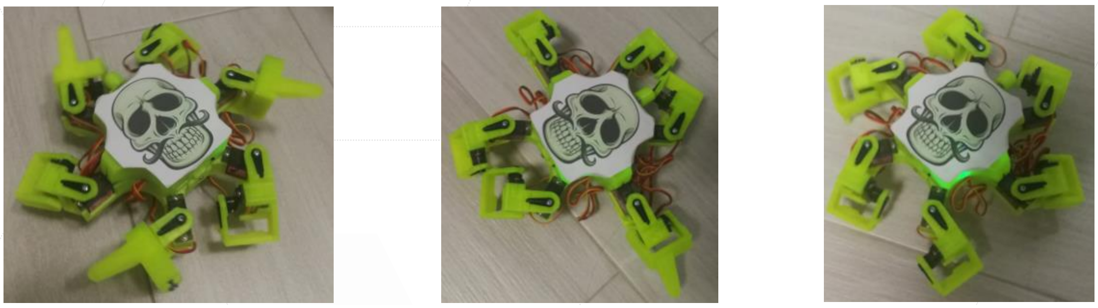
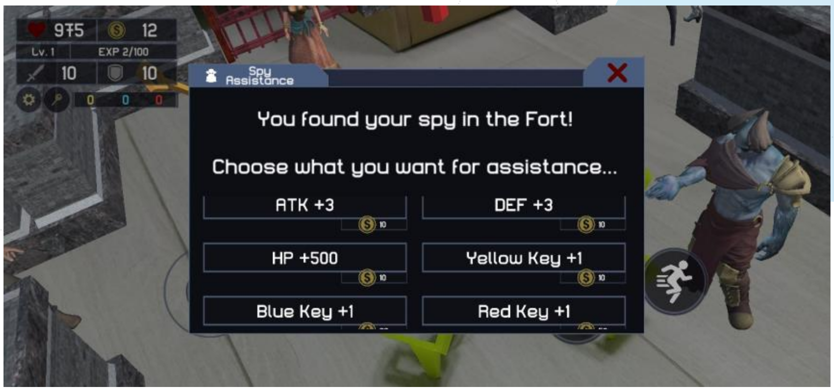


Figure 37: Example of celebration actions

3.11.5 Shop System

The shop system is existed in the game in the form of a spy that was sent by the players to the fort, so as to provide assistance with rewards needed for the spy (see Figure 38).



You found your spy in the Fort!

Choose what you want for assistance...

Figure 38: Spy assistance window

The money players gained from beating the enemies can used to buy different assistances, such as increasing the ability of the player character or buying extra keys provided by the spy. However, the appearance of the spy is similar to that of the enemies (see Figure 39), the shop system will only trigger when player collide with the spy. Therefore, players can only guess and try by luck if they want to access to the shop system.

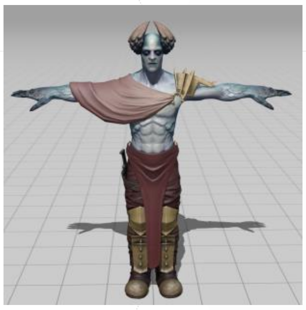
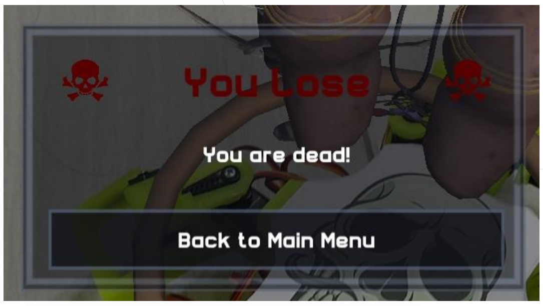


Figure 39: Model of spy

3.11.6 Ending Condition

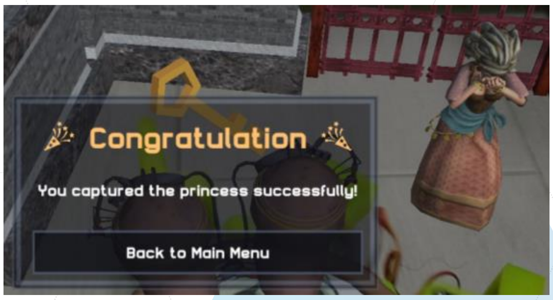
Players are lost if they dead during the fighting with enemies before capture the princess successfully (see Figure 40). Otherwise, they win when they capture the princess by colliding with the princess (see Figure 41). However, the appearance of the princess is similar to that of the enemies and spy too (see Figure 42), so again, players have to guess which character is the real princess. Upon winning, one of the celebration actions mentioned above will be randomly selected and performed automatically by hexapod.


You Lose

You are dead!

Back to Main Menu

Figure 40: Pop-up when lost


Congratulation

You captured the princess successfully!

Back to Main Menu

Figure 41: Pop-up when won

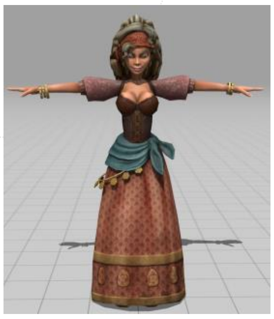


Figure 42: Model of princess

3.12 Automatic Saving and Loading in Local

The automatic saving of game progress is applied every time when the players performed any actions that affect the progress, such as picked up an item, opened a door or fought with an enemy. The record is stored locally. In order to save the progress, Hexplore Fort should store all the player's statistic and progress such as which area have the player unlocked previously.

Since the data structure of the record needed to be stored is too complicated for **PlayerPrefs**, which is storing the data in **SharedPreferences** for Android [19]. Since **PlayerPrefs** only supports saving of primitive data type, such as integer, float and string, writing the record in form of serializable class into a binary data file and storing in local storage is used instead.

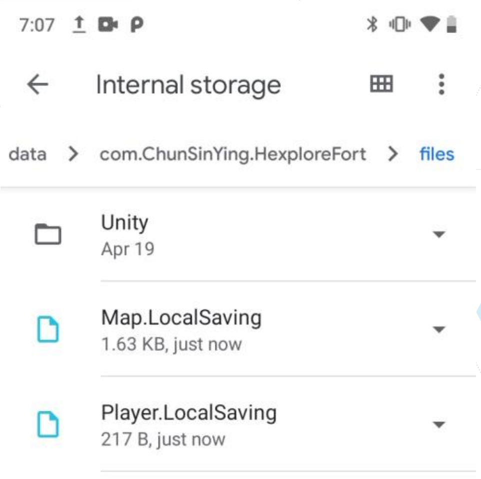
To save the record, a file is created in the specified path.

```
BinaryFormatter formatter = new BinaryFormatter();
FileStream stream = new FileStream(path, FileMode.Create);
formatter.Serialize(stream, objectOfSerializableClass);
```

To load the record, a file is opened in the specified path.

```
BinaryFormatter formatter = new BinaryFormatter();
FileStream stream = new FileStream(path, FileMode.Open);
SerializableClass object = (SerializableClass)formatter.Deserialize(stream);
```

The binary data file is stored in local file system of Android (see Figure 43).



Unity
Apr 19

Map.LocalSaving
1.63 KB, just now

Player.LocalSaving
217 B, just now

Figure 43: Binary data files saved and loaded by Hexplore Fort

3.13 Settings

There are several settings provided to the players to support the players if they encountered with any difficulties (see Figure 44). For example, they can tune the volume of the audio played in the game. Besides, they can reconnect the hexapod, retrack all the AR elements and exit the game in the mid-way of the gameplay. The gameplay will be paused if the below setting window is active.

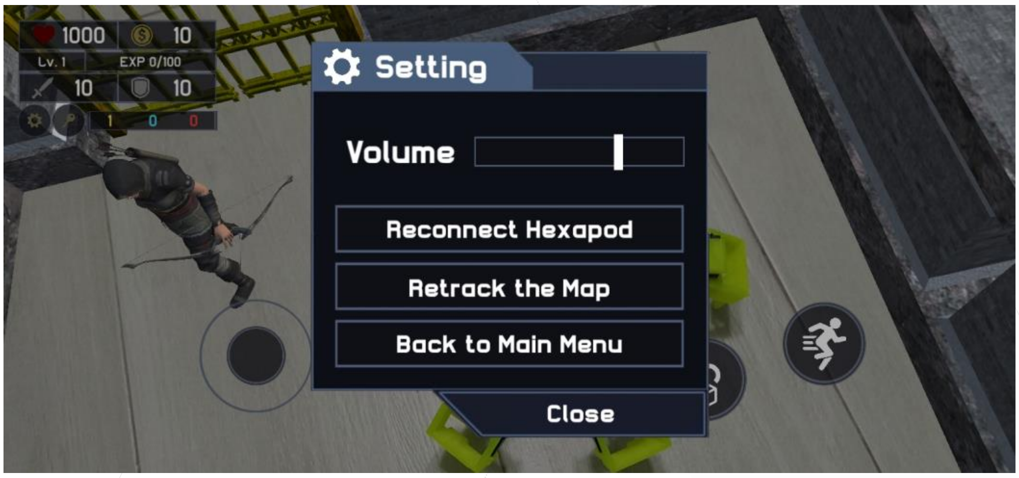


Figure 44: Setting window

3.13.1 Volume Tuning

Players can tune the volume of the background music and sound effect by the slider provided.

3.13.2 Reconnection of Hexapod

Players can reconnect the hexapod if the robot is disconnected accidentally. After reconnection, the game will continue at where the players stop.

3.13.3 Retracking of AR Elements

Players can retrack all the AR elements if they encountered with any problem of lost track to the robot or the map that cannot be easily resume the tracking by point the camera at the anchor point; or if the players wish to place the map into another place that they want. This is done by destroying the ARCore session and creating a new ARCore session. The automatic saved record will be loaded to the new session.

3.13.4 Exit Game

Players can back to the main menu in the mid-way of the gameplay. Since the game record is automatically saved, players can continue their game by pressing "Continue" button in the start menu.

3.14 Player Game Testing

After testing, the game testers have found several potential improvements for the game to make it more user friendly. For example, the testers have reported that when picking up items, the indication for the players that an item has been picked up was not very clear. The original item pickup indication for the player were the disappearance of the item as well as the item count going up. However, the item disappearance is largely blocked by the physical robot as the robot takes up a considerable amount of screen space. The item count was not very visible as well. Therefore, a pop-up notification has been added to notify the user that an item has been picked up more clearly (see Figure 45).

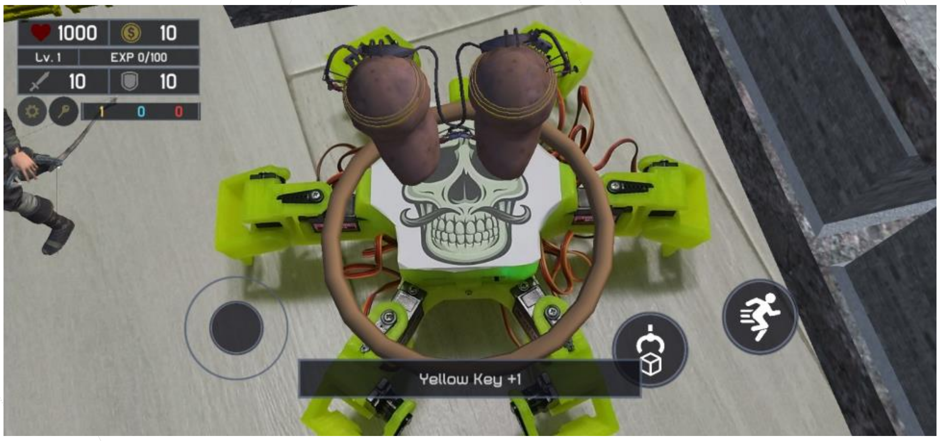


Figure 45: Pop-up notification when item picked up

The testers have also reported that the outlook of the virtual player augmented on top of hexapod was not fit for the theme of the game. Therefore, the appearance of the virtual player has been changed to fit more into the theme of the game (see Figure 46).

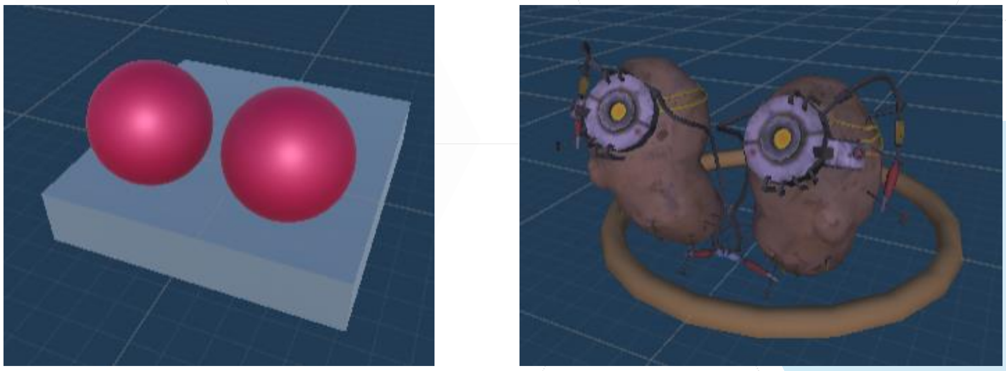


Figure 46: Model of virtual player; before (left), after (right)

The testers have also mentioned about the difficulty ramp of Hexplore Fort over time was not very smooth. Some of the fights are either too long, or overly difficult. Therefore, the statistics of the enemies were tweaked, so that the enemy's difficulty is more in line with the player character's growth throughout the game.

4 Limitations and Difficulties Encountered

Since the map augmented is virtual, while the hexapod used to walk on the map is physical, this caused some difficulties on preventing the players cheating for the gameplay easily.

First, the problem of hexapod walking through the virtual wall augmented by AR is encountered because the hexapod is physical and is not blocked by any physical obstacles. This problem is solved by augmenting a virtual player on top of the hexapod, and detecting the collisions happened in the front and at the back of the virtual player (see Figure 47). When either one of the colliders shown in the figure is collided with the virtual wall, the game will block the sending of Bluetooth movement control command to the direction of the wall.

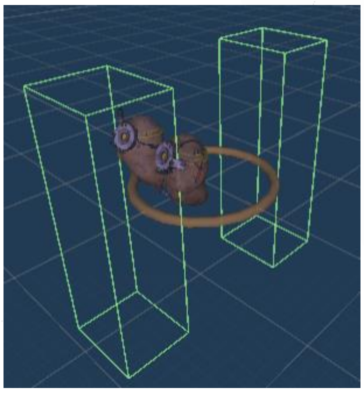
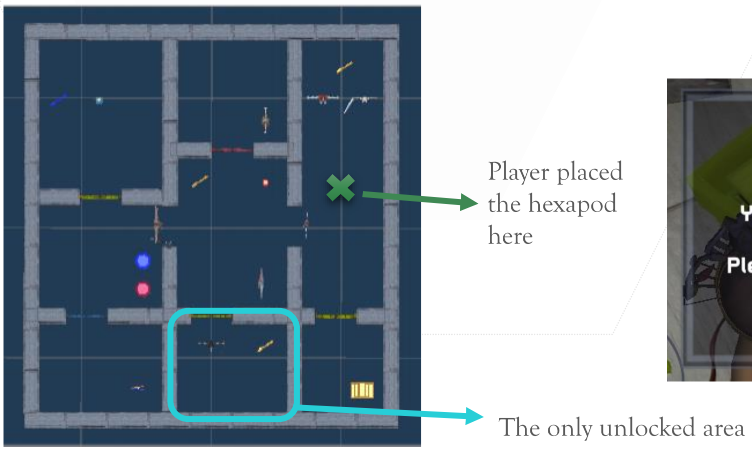
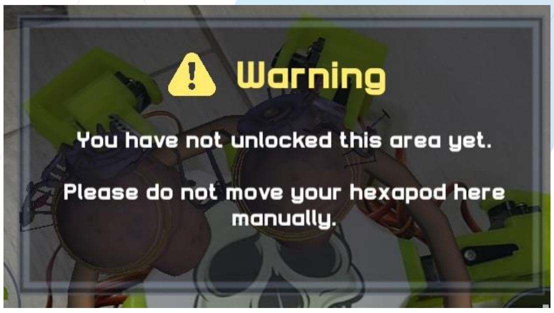


Figure 47: Virtual player with front and back colliders visible

Second, the problem of "teleportation" is caused by the availability of players placing the hexapod in any place of the map augmented. This problem might break the gameplay because the players can place the hexapod to capture the princess directly. The current solution is storing the data of area that the players have unlocked, and disable all game features like moving, fighting and shopping when it is detected that the players reached the locked area (see Figure 48).





Warning

You have not unlocked this area yet.

Please do not move your hexapod here manually.

Figure 48: Example of forbidden "teleportation"

Other than the above solved difficulties, some limitations of using ARCore for recognizing the robot was found.

Since object recognition is not available in ARCore, image recognition is used as an alternative. However, the robot body is in plain color and without any characteristic features, this increases the difficulty of recognizing the robot by the AR camera. Therefore, in order to recognize the robot, a reference image is needed to be stuck on the body of the robot., unless an image with many characteristic features is stuck on the robot body.

Moreover, using image recognition has several requirements as stated in the official documentation of ARCore. The images must be flat, in clear view of the camera and fill at least 25% of the camera frame to be initially detected [18]. In other words, the camera should always be close to the hexapod in order to recognize the moving of the hexapod, otherwise, the virtual player augmented will only stay on the same place where the last tracking was (see Figure 49).

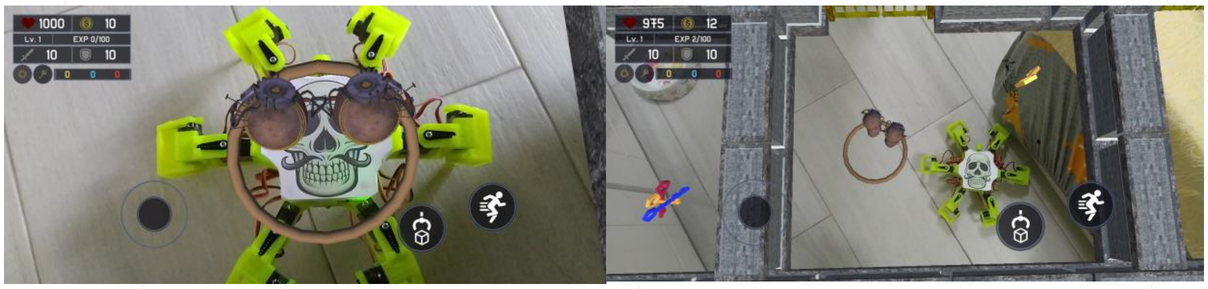


Figure 49: Example of image tracking; close-up (left), far away (right)

Apart from it might be unfriendly to user experience for maintaining the hexapod always under the camera, it might sometimes cause the lost track of detected plane, and the map will not be stable at a fixed position as a result. Since it is mentioned that ARCore detect plane by looking for clusters of feature points which appear to lie on common horizontal flat surface, if the camera is too close to the hexapod where the hexapod occupied the most of the camera frame, the camera may lost track of the feature points tracked for the plane. This cannot be solved since it is the limitation of image tracking of ARCore, therefore, a button for players to manually retrack the AR elements is the only alternative to maintain the gameplay after problems occurred.

5 Future Development

Since Hexplore Fort developed for this project is mainly for demonstrating the combination of physical and virtual gaming experience, the level system is eliminated, and Hexplore Fort is having only one fixed map. Therefore, level system can be left as future development. Taking reference to Tower of the Sorcerer, it can have more than hundreds of floors with many kinds of enemies. This can be a long-term development for endless fort exploration.

The enemies in Hexplore Fort are not equipped with any special skills, it can be one of the development directions too.

As times goes on, if object detection is available in ARCore, the new AR technology can be used to replace the image recognition of robot in used currently. This will greatly reduce the limitation of the game caused by ARCore.

6 Conclusion

The goal of this project is to provide a better gaming experience to players by blending physical and virtual gaming experience. In order to demonstrate the combination of the two, a single-player RPG adventure AR Android mobile game called Hexplore Fort which supports the control of robots to do the adventure with the storyline is delivered.

The project is done by developing a robot controlling application by Android Studio with Java first, in order to control the 3D-printed Arduino-based hexapod. It is then transformed into a plugin to be used in Unity. The gaming environment is built by using Unity with C#, ARCore SDK is integrated into Unity for use.

The virtual terrain is augmented onto the ground by using plane detection of ARCore, while the robot recognition is done by using image tracking.

The game Tower of the Sorcerer is highly referenced for the game design and mechanism. Players are required to control the movement of the hexapod for exploration in the augmented virtual terrain, and performed actions like collecting items, combating with enemies and receiving assistance form the spy etc.

Although some technical difficulties are encountered, alternatives are completed to prevent players from cheating. However, some limitations caused by using the image recognition of ARCore still existed, and alternatives are provided after the problem is occurred but not to prevent the problem to occurred.

Future development is suggested to follow the direction of the whole game flow in Tower of the Sorcerer. Replacing the image recognition used in this project by new AR technology if there is any updates from ARCore is also recommended.

The project is completed on schedule with the scope as same as stated in the proposal

References and Attributions

[1] Limelight Networks. 2019. “The State of Online Gaming 2019.” Accessed September 28, 2019. https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019/.

[2] Newzoo. 2019. “Betting on Billions: Unlocking the Power of Mobile Gamers.” Accessed September 28, 2019. https://cdn2.hubspot.net/hubfs/4963442/Whitepapers/ABM_Newzoo_Betting_on_Billions_Unlocking_Power_of_Mobile_Gamers_March2019.pdf.

[3] Statista. September 16, 2019. “Top grossing iOS mobile gaming apps 2019, ranked by daily revenue (in U.S. dollars).” Accessed September 28, 2019. https://www.statista.com/statistics/263988/top-grossing-mobile-ios-gaming-apps-ranked-by-daily-revenue/.

[4] PW - Pokemon Go by Virginia State Parks is used under a CC-BY 2.0 license. Retrieved from Flickr https://www.flickr.com/photos/vastateparksstaff/27972275293.

[5] Pokemon Go by Masaki Tokutomi is used under a CC-BY 2.0 license. Retrieved from Flickr https://www.flickr.com/photos/tokutomi/32130329993.

[6] MarketWatch. August 28, 2019. “Entertainment Robots Market Is expected to reach USD 2,505.12 million by 2023.” Accessed September 28, 2019. https://www.marketwatch.com/press-release/entertainment-robots-market-is-expected-to-reach-usd-250512-million-by-2023-2019-08-28.

[7] Boston Dynamics. Accessed September 28, 2019. https://www.bostondynamics.com/about.

[8] BostonDynamics. “More Parkour Atlas.” YouTube video, September 24, 2019. https://www.youtube.com/watch?v=_sBBaNYex3E.

[9] Fandom. “Tower of the Sorcerer.” Accessed September 28, 2019. https://tig.fandom.com/wiki/Tower_of_the_Sorcerer.

[10] Trossen Robotics. “PhantomX AX Metal Hexapod Mark III Kit.” Accessed September 28, 2019. https://www.trossenrobotics.com/phantomx-ax-hexapod.aspx.

[11] Vorpal. “Vorpal The Hexapod Gamepad User Guide.” Accessed September 28, 2019. https://vorpalrobotics.com/wiki/index.php/Vorpal_The_Hexapod_Gamepad_User_Guide.

[12] Github. April 6, 2020. “BugBot” Accessed April 15, 2020. https://github.com/hkucs-makerlab/BugBot

[13] Travel by victor217 is used under a CC-BY 2.0 license. Retrieved from freepik
https://www.freepik.com/free-photos-vectors/travel

[14] Potato by Ru is used under a CC-BY 2.0 license. Retrieved from Poly
https://poly.google.com/view/23xm2GMCkqn

[15] Github. December 9, 2017. “HTML5 魔塔樣板” Accessed March 10, 2020.
https://github.com/ckcz123/mota-js

[16] Github. February 19, 2018. “mota” Accessed March 13, 2020.
https://github.com/justinmzt/mota

[17] ARCore. April 15, 2019. “Fundamental concepts” Accessed February 24, 2020.
https://developers.google.com/ar/discover/concepts

[18] ARCore. March 19, 2019. “Recognize and Augment Images” Accessed February 25, 2020.
https://developers.google.com/ar/develop/unity/augmented-images

[19] Unity. April 29, 2019. “PlayerPrefs” Accessed March 8, 2020.
https://docs.unity3d.com/ScriptReference/PlayerPrefs.html