

Department of Computer Science and Engineering  
The Chinese University of Hong Kong  
ESTR4998 Graduation Thesis Report

LYU1904

# **College War - An AR mobile game**

Written by

Zhao Haodong(1155092149, Hardy)

Li Jiazhou(1155092117, Daniel)

Supervised by

Prof. Lyu Rung Tsong Michael

25 May 2020

<b>Abstract .....</b>	<b>5</b>
<b>Acknowledgements.....</b>	<b>6</b>
<b>1. Introduction .....</b>	<b>7</b>
1.1. Motivation.....	7
1.2. Background .....	8
1.2.1. Unity .....	8
1.2.2. Augmented reality .....	8
1.3. Objective.....	9
<b>2. Design.....</b>	<b>10</b>
2.1. Game Overview .....	10
2.2. Gameplay .....	11
2.3. Game Components.....	13
2.3.1. Map.....	13
2.3.2. College.....	14
2.3.3. Collect system.....	14
2.3.4. Illustrated handbook.....	15
2.3.5. Attack and Defense System.....	16
2.3.6. Attack system.....	16
2.3.7. Defense system.....	17
<b>3. Implementation .....</b>	<b>19</b>
3.1. Map.....	19
3.1.1. Overview .....	19

3.1.2.	OSM file analyzation .....	19
3.1.3.	Mercator Projection .....	21
3.1.4.	Different types of region.....	22
3.1.5.	Building .....	22
3.1.6.	School Bus Route .....	23
3.1.7.	Road.....	24
3.1.8.	Others.....	24
3.2.	Attack and Defense System.....	25
3.2.1.	UI Design .....	25
3.2.2.	Attack weapons .....	28
3.2.3.	Defense Installations .....	30
3.3.	Collect System.....	32
3.3.1.	UI Design .....	32
3.3.2.	Basic Materials.....	33
3.3.3.	Inventory and Shop .....	35
<b>4.</b>	<b>Research in AR.....</b>	<b>37</b>
4.1.	Overview .....	37
4.2.	Mobile AR Development Framework .....	37
4.2.1.	Overview .....	37
4.2.2.	Vuforia Engine .....	39
4.2.3.	Image Recognition .....	40
4.2.4.	3D object recognition .....	41
4.2.5.	Advantages and disadvantages.....	42
4.2.6.	ARCore.....	42
4.2.7.	Motion tracking.....	42

4.2.8.	Environment understanding .....	44
4.2.9.	Choose of tool .....	45
4.3.	Point Cloud for Classification .....	45
4.3.1.	Introduction.....	45
4.3.2.	Related work.....	46
4.3.3.	Network structure .....	49
4.3.4.	Objective function .....	50
4.3.5.	Classification algorithm .....	50
4.3.6.	Result and Discussion .....	51
4.3.7.	Conclusion.....	53
<b>5.</b>	<b>Network .....</b>	<b>54</b>
5.1.	Overview .....	54
5.2.	Server.....	54
5.2.1.	Connection Listener .....	55
5.2.2.	Game Logic Processor .....	55
5.2.3.	Database .....	55
5.2.4.	Detailed implement of functions .....	57
<b>6.</b>	<b>Conclusion .....</b>	<b>62</b>
6.1.	Summary of second term .....	62
<b>7.</b>	<b>Division of labor .....</b>	<b>63</b>
	<b>Bibliography .....</b>	<b>64</b>

# Abstract

In this project, we focus on designing and developing an online mobile game with augmented reality(AR) technologies and elements derived from CUHK Campus. We would apply the map of our campus as well as the college system into the game. AR games have certain advantages that they act as new ways of expression and give players the sense of interaction in the real world. Moreover, unlike virtual reality(VR) games which need players wear equipment, AR games have effects that can be shown in mobile phones very well. So that we integrate our game with AR technologies. In this term, we only implement an offline version of the game with the framework. However, we hope that after the full implementation in next semester, the game may encourage students and stuffs of CUHK to explore the beautiful campus more.

# Acknowledgements

We would like to express our greatest gratitude to our supervisor Prof. Michael Lyu and advisor Mr. Edward Yau for providing guidance, feedback and resources in this interesting project.

Besides, we are also very appreciated to everyone who has supported us in the project or read this report.

# 1. Introduction

## 1.1. Motivation

*“One day, we believe this kind of immersive, augmented reality will become a part of daily life for billions of people.”*

*—Mark Zuckerberg, Founder and CEO of Facebook Inc.*

During the recent few years, mobile technologies have diffused into all consumer segment (1). AR provides opportunities for various industries, including those that develop interactive and engrossing game (2). For example, Goldman Sachs indicated in 2016 report that AR has the potential to “become the next big computing platform, and as we saw with the PC and smartphone, we expect new markets to be created and existing markets to be disrupted”. There is no doubt that AR will new the existing game development. We would like to design a game with the CUHK campus as the map background. Since our campus is a very beautiful campus built on the mountain, we hope our game can give players a chance to better explore our campus. Moreover, while keeping the basic design of the reference game, we would like to add elements of augmented reality to give players more novel and interesting experience. Furthermore, we want to introduce the college system of CUHK into our game by dividing players into different groups and making them fight for their own group and college which can increase the players’ sense of substitution and belonging. At the same time, we hope that such a design can attract more players and that’s why we named the game CollegeWar.

## 1.2. Background

### 1.2.1. Unity

Unity is a very popular and powerful game engine that supports both virtual reality and augmented reality games. While bring convenience to our game development, unity offers packages and supports third party packages that could help us with AR technology.

### 1.2.2. Augmented reality

Both of virtual reality(VR) and augmented reality(AR) are becoming more and more popular. There are many VR trial spots in large spots as well as many applications in app stores. While VR allows users immerse in the three-dimensional virtual environment generated by computer and be separated from the real world, AR adds or removes real-time interactive objects and information generated by computer in real environment. One of the disadvantages of virtual reality is that users must wear the corresponding equipment. On contrary, users can use their mobile phones to experience the AR effects so that we choose to integrate augmented reality into our mobile games.

The use of AR in games is particularly popular. A very good example is Pokémon go, which have attracted the interest of worldwide players. Media named the Pokémon Go “the biggest mobile fame in U.S. history” (3). As a new way of game expression, AR game has certain advantages. It gives players the sense of interaction in the real world and fully mobilizes their enthusiasm and initiative in playing games. Moreover, using AR technology makes the game more interesting and diversified.

However, mobile AR as a platform can generally only offer isolated, solitary, experiences that will seldom result in the kind of critical use cases. Recently, with 5G technology development, how to combine AR with attributes of 5G, such as

greater bandwidth and faster download speed, becomes one of the research directions. AR Cloud technology, the framework for information and experience augment and sharing, has been proposed for blending digital and physical, which will promote the development of relevant applications.

### **1.3. Objective**

The goal of the entire project is to design and build a mobile game with CUHK as its background. It should be location based and contains campus system as well as elements of AR. Last but not the least, the game will be an online game. We hope, via the release of the game, both students and staffs of CUHK will like it and get the chances to explore our campus more.

The entire project is expected to be completed in 2 terms. Last term we designed and implemented an offline version of the game. We rendered CUHK map in the game with data collected from the internet and implemented AR scenes with Vuforia. Here are the objectives for the second term:

- Complete network module and integrate it with our game.
- Enrich collection system and combat system with more specific details.
- Optimize map rendering with different types of regions and make it more clear and friendly to players.
- Update UI and replace resource materials.

## 2. Design

### 2.1. Game Overview

The name of our game is CollegeWar, which is an RPG and SLG game of college battle with Android as its platform. It is a location-based, multi-player online game, with CUHK campus as its main map, with AR as its crucial property and targeting only for students and staffs in CUHK.

The first-time players enter CollegeWar, they are asked to choose one college which can't be changed later. To play the game, players need to physically be at CUHK campus. Then by walking around our campus, players may get chances to collect different kinds of materials. With these materials, players can buy weapons and defense installations which can be used to attack or defense buildings in shop menu. As the more buildings players occupied, the more materials they could gain producing by these buildings.

As stated by John Hopson via Gamasutra in 2001, a forced circulation in game design is very important. We also introduced it into our game. As the main goal for players is to attack buildings. To achieve this, they need to collect materials and buy weapons or defense installations. At the same time, the more buildings players occupying, the more materials they could get from them. In this way, players can attack and occupy much more buildings. The whole circulation is interconnected and interactive which can let players play our game spontaneously and voluntarily.

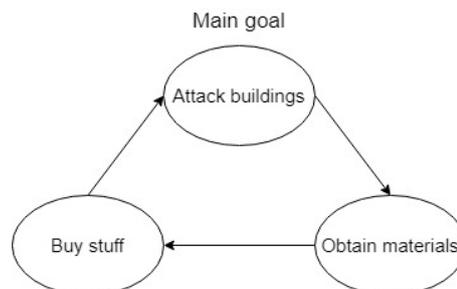


Figure 2-1 Forced circulation

## 2.2. Gameplay

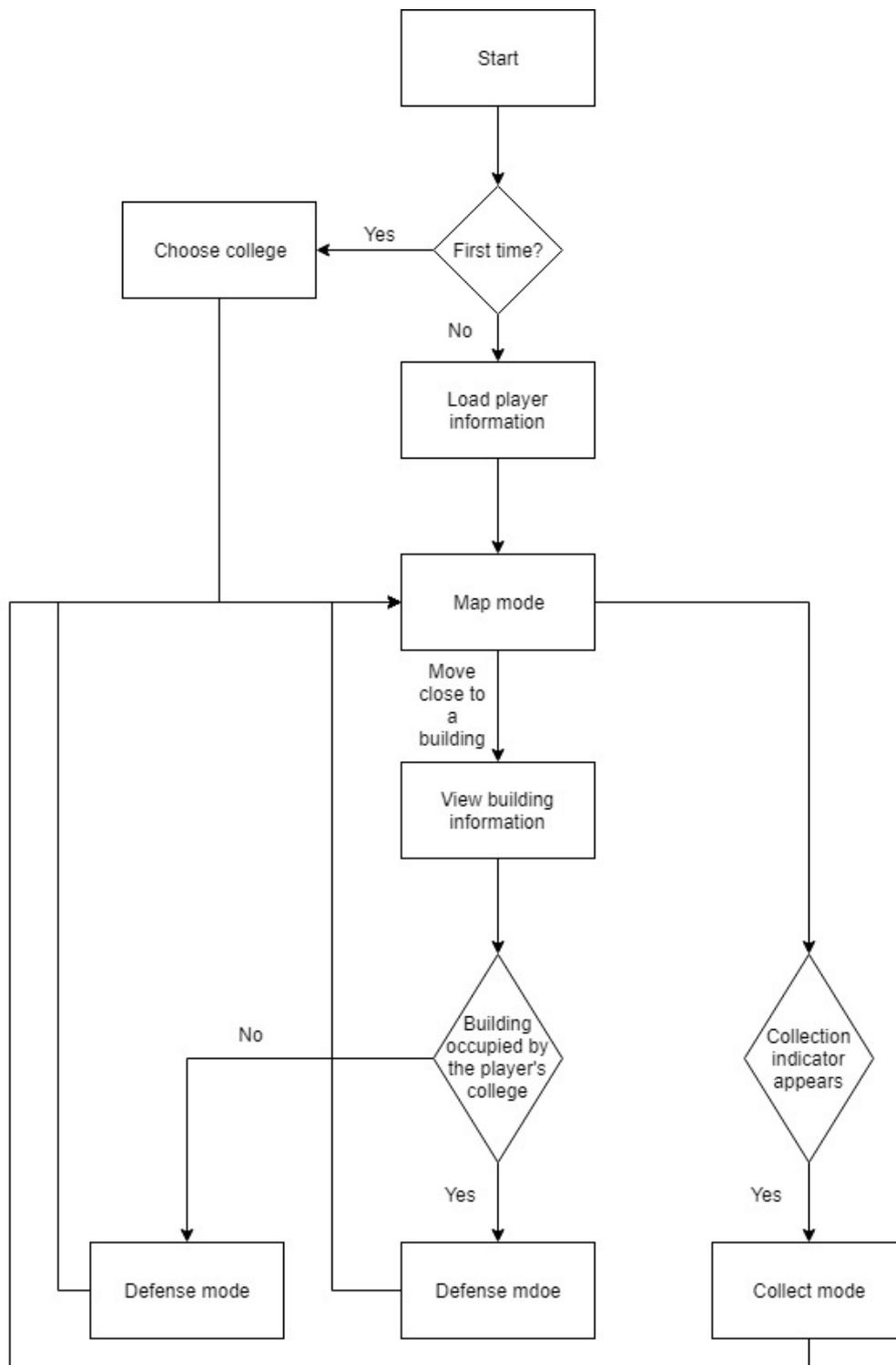


Figure 2-2 Block diagram of gameplay

The first-time players start the game, they are asked to choose their own colleges. Once chosen, players can't change it. After chosen, an avatar which indicates the player is displayed on a map based on the player's geographical location. This is called map mode.

Main features on the map are buildings including neutral ones and occupied ones as well as different types of regions. The buildings are typically the same as buildings in physical world in shapes and locations. A building is neutral means it is not occupied by any colleges. On contrast, a building is occupied means it is currently occupied by a college. An occupied building produces materials for players belonging the college that occupies the building. The different map regions are generated base on the real-world environment including grass, water, road and playground.

As players move within their real-world surroundings, their avatars move within the game's map. An indicator on UI appears if there are collectible materials around players. Indicators of different icons means different types of materials. Then players may enter collect mode to collect materials.

If a player wants to attack or defense a building, he/she needs to physically move to the building before entering attack or defense mode. Players can attack buildings that are not occupied by its college and defense buildings that are occupied by its college.

## 2.3. Game Components

### 2.3.1. Map

Our game is a location-based game with CUHK Campus as its map. It means that to play the game, players need to be physically at the campus. As players move within their real-world surroundings, their avatars move within the game's map. Besides, the map is a 3D map that contains buildings and different regions of CUHK. The buildings in the map are generally the same with the buildings in real world in names, shapes and locations. Moreover, different regions including grass, water, road and playground in the map are also generated based on real world.

In map mode, players can view inventory information and buy materials in shop menu. By double clicking on buildings in the map, players can view the detail information and choose to attack or defense them. Moreover, since our game is a multi-player online game, other players who are also playing the game will be shown in the map with position being updated.

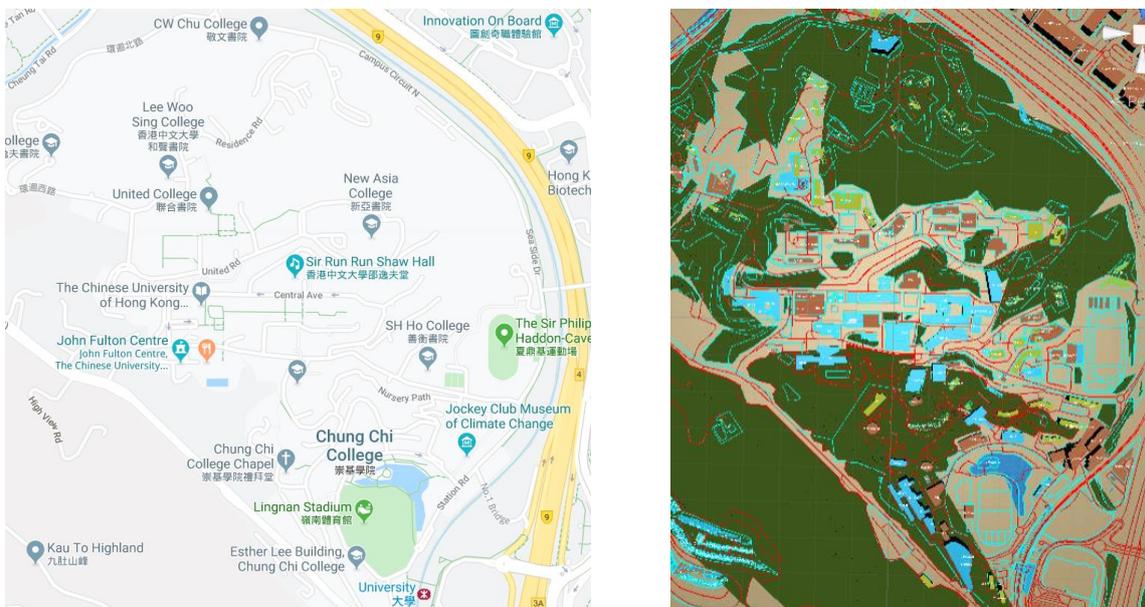


Figure 2-3 Google map and rendered map of CUHK in the game

### **2.3.2. College**

We introduced 9 colleges in CUHK into our game. The main reason for doing this is because we want to take advantage from players' sense of belonging to their own college. It's very likely for them to be proud if their colleges are very powerful and occupying biggest number of buildings in the game. In this way, they'll have more motivation in playing the game. It's also one of the characteristics we would like to use to attract users.

Each college has few buildings that are unattackable by other players such as its dormitory buildings. The game requires players to choose a college when they first start it. Once chosen, it's unchangeable. In this term, we don't require players to choose the college as they belong to in the real world. If possible, in the future, we would like to restrict players to choose colleges they belong to in some ways. Considering that the number of players from different colleges may vary a lot, we have come up with several solutions to it. For example, compensate colleges with fewer players or add some machine players.

### **2.3.3. Collect system**

To buy weapons and defense, players need to collect materials in collect mode. There are four kinds of materials including wood material, fish material, road material and power material. An indicator on UI with different icons shows which kind of materials that players are able to collect materials. The type of the material is decided by the real-world surroundings of the player which means players need to physically walk close to regions with grass, water, road or playground to collect corresponding materials.

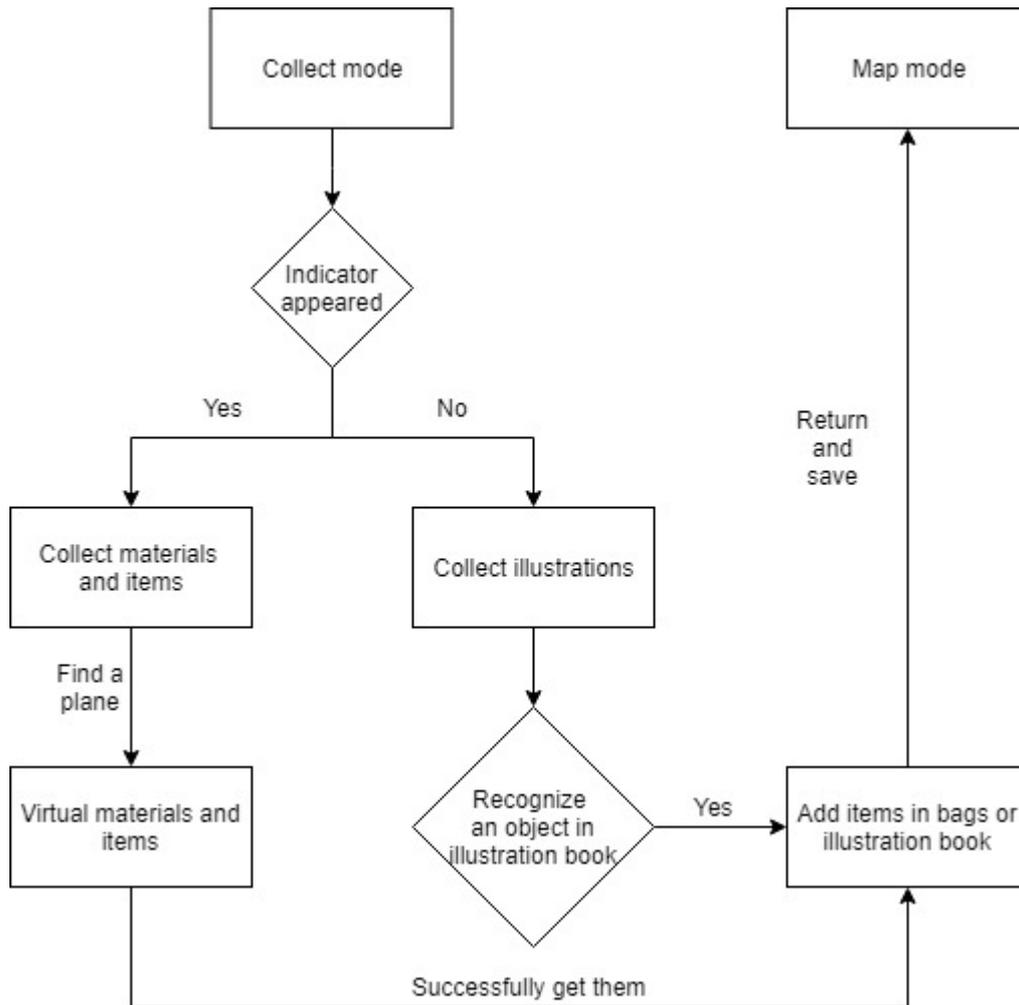


Figure 2-4 Block diagram of collect system

#### 2.3.4. Illustrated handbook

In order to enhance player experience except from interest in game, more elements on map for exploration provides extra target for players to go outdoors. Our game provides players with an illustrated handbook to record the collected items. The handbook contains all the materials weapons and defense. In addition, popular signs of CUHK, such as the pool in New Asia College and the female foot model in Shaw College, are designed for tourist and freshmen keeping better knowledge about CUHK campus.

Through this, we encourage players to explore CUHK campus and record the attractive moments. We also want to see players are surprised at finding their surroundings are shown in the game. At the same time, players are encouraged to share information with each other. Players can obtain returns according to the number of unlocking stamps of the handbook so that they will be likely to do it actively.

### **2.3.5. Attack and Defense System**

In our game, the main parts are the attack and defense system. Because players are divided into 9 colleges and everyone fights for their own college, how to occupy more buildings in CUHK is the target of players. In the beginning of the game, all the buildings are natural and will be occupied by colleges later. The motivation for players to attack buildings is that the more buildings the players own, the more materials they can get from them. With more materials, they can buy more weapons to attack more buildings or defense their buildings better.

### **2.3.6. Attack system**

As the main goal of players is to attack and occupy buildings, the attack system is very important. Players can buy weapons in the shop which costs materials. Materials are gained by collecting and produced by buildings that the player's college occupies. We have designed two weapons which are arrow and bomb. They have different characteristics and require players to operate in different ways. During attack mode, players can view the building and other players who are currently attacking or defending it in AR environment.

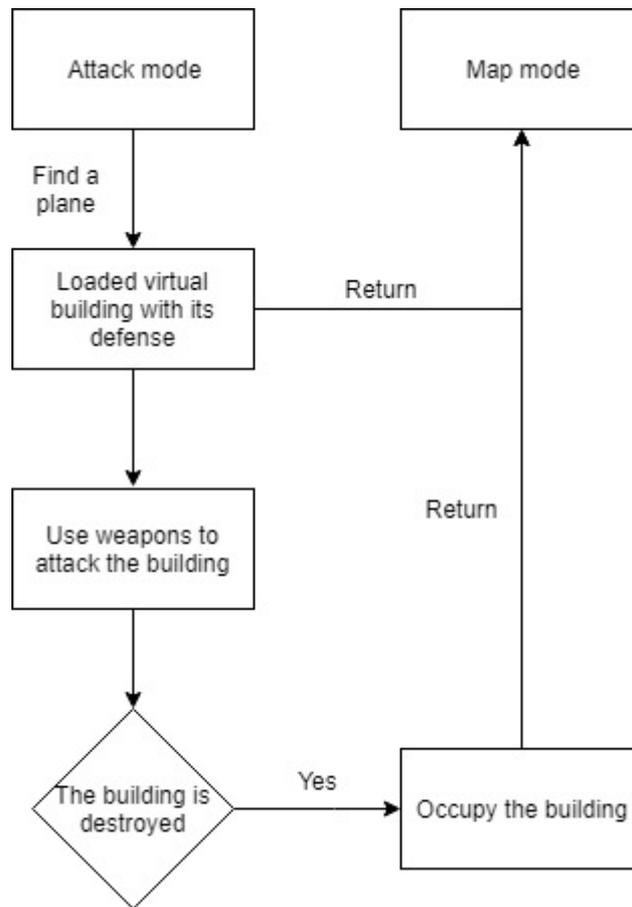


Figure 2-5 Block diagram of attack system

### 2.3.7. Defense system

In defense mode, players may add defense installations including walls, shields and alarms and adjust their positions to better defend buildings and to avoid buildings being attacked and occupied by other colleges. The benefit of occupying and defending buildings is that for each hour, all the occupied buildings will produce materials to players belonging to the college.

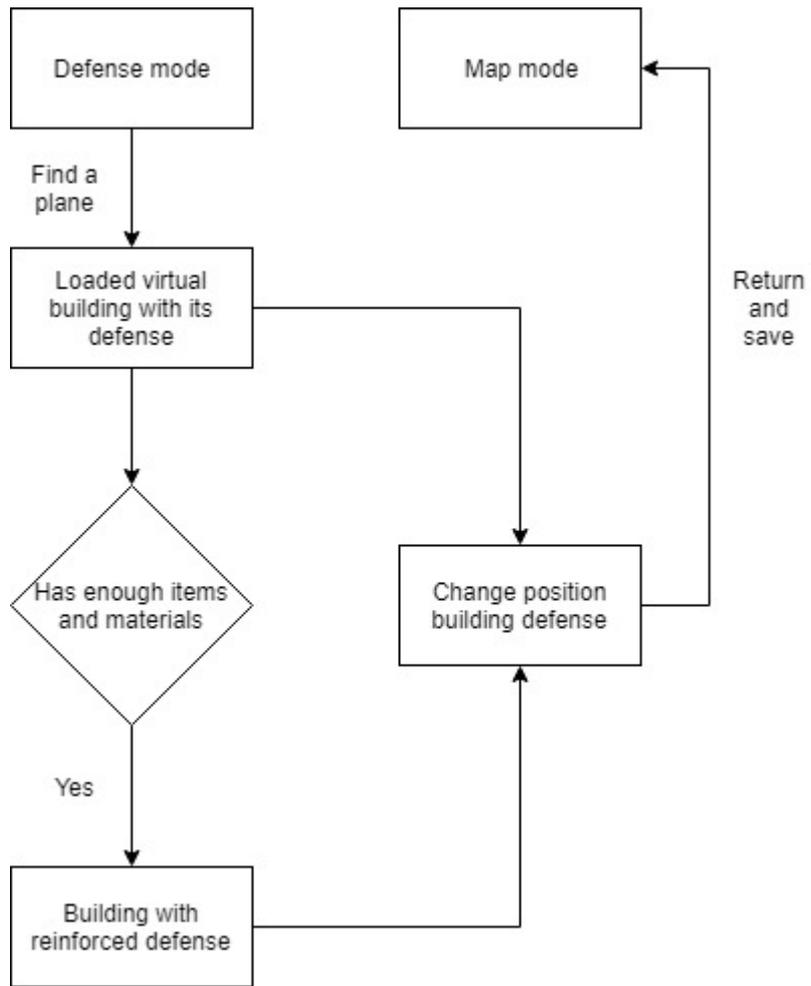


Figure 2-6 Block diagram of defense system

# 3. Implementation

## 3.1. Map

### 3.1.1. Overview

In order to provide players with more reality experience, how to restore the CUHK map, with building, facility, road, and some natural region distributions is one of our goal. Comparing current online open source map dataset, we choose the OpenMapStreet, an open source map data website. It is a collaborative project for creating editable map of the world, with the geodata distributed by multiple volunteers, ground survey mappers, and government agencies. By accessing the raw map data of CUHK, attributes of basic facility are convenient for analysis, such as GPS position, name with Chinese and English version, land use, usage type, and other specifications. According to the analyzation, the map can be generated more detailed and abundant.

### 3.1.2. OSM file analyzation

OpenMapStreet provides multiple export file format for developer to choose. Comparing with more compact or simpler encoding type, we choose the original XML-based OSM format. This format is rater verbose but provides detailed information on region GPS description, which is helpful for 3D object rendering. In addition, because XML is a so called meta format, the clear structure provides better human readable experience.

The XML suffix introducing the UTD-8 character encoding for the file, and the basic element types consists of node, way and relation.

- Node

Its main function is to especially provide the location in the WGS84 reference system. Its attribute only contains tags with key-value pair for description.

```
<node id="1486192772" visible="true" version="9" changeset="36087448" timestamp="2015-12-21T14:37:54Z" user="Mapier" uid="616884" lat="22.4194681" lon="114.2053810">
  <tag k="layer" v="2"/>
  <tag k="name" v="仲門 The Gate"/>
  <tag k="name:en" v="The Gate"/>
  <tag k="name:zh" v="仲門"/>
  <tag k="region" v="Main Campus"/>
  <tag k="tourism" v="artwork"/>
  <tag k="type" v="Landmark"/>
</node>
```

Figure 3-1 Node attributes

- Way

It is the collection of nodes, mainly for represent a region with polygons, such as building, wood area, water region, and lines of road. A way can have between 2 to 2000 nodes, closed if the first node is equal to the last node id. Its attribute contains the references to its nodes, and the tags with key-value pair for description, e.g. region type, land-use, name, building levels and some comments.

```
<way id="135312912" visible="true" version="9" changeset="71141536" timestamp="2019-06-11T13:23:09Z" user="cheungcm" uid="10054182">
  <nd ref="1486165682"/>
  <nd ref="1486165579"/>
  <nd ref="1486165574"/>
  <nd ref="1486165680"/>
  <nd ref="1486165687"/>
  <nd ref="1486165595"/>
  <nd ref="2389077297"/>
  <nd ref="1486165676"/>
  <nd ref="2389077301"/>
  <nd ref="1486165682"/>
  <nd ref="1486165660"/>
  <nd ref="1486165613"/>
  <nd ref="1486165608"/>
  <nd ref="1486165589"/>
  <nd ref="1486165682"/>
  <tag k="building" v="university"/>
  <tag k="building:levels" v="10"/>
  <tag k="name" v="何善街工程學大樓 Ho Sin-Hang Engineering Building"/>
  <tag k="name:en" v="Ho Sin-Hang Engineering Building"/>
  <tag k="name:zh" v="何善街工程學大樓"/>
  <tag k="region" v="Main Campus"/>
</way>
```

Figure 3-2 Way attributes

- **Relation**

It is a group of elements. To be more exact speaking, relation is one of the core data elements that consists of one or more tags and also an ordered list of one or more nodes, ways and/or relations as members which is used to define logical or geographic relationships between other elements. Its attribute contains the references to its nodes, ways and relations, the members with relevant relation description, and the tags containing attribute description.

```
<relation id="7802779" visible="true" version="6" changeset="74492891" timestamp="2019-09-15T10:18:09Z">
  <member type="way" ref="33277733" role="outer"/>
  <member type="way" ref="546126973" role="inner"/>
  <member type="way" ref="546626972" role="outer"/>
  <member type="way" ref="546626971" role="outer"/>
  <member type="way" ref="185266976" role="inner"/>
  <member type="way" ref="724693008" role="outer"/>
  <tag k="amenity" v="university"/>
  <tag k="name" v="香港中文大學 The Chinese University of Hong Kong"/>
  <tag k="name:de" v="Chinesische Universität Hongkong"/>
  <tag k="name:en" v="The Chinese University of Hong Kong"/>
  <tag k="name:lt" v="Honkongo kiniškasis universitetas"/>
  <tag k="name:ru" v="Китайский университет Гонконга"/>
  <tag k="name:zh" v="香港中文大學"/>
  <tag k="name:zh-Hans" v="香港中文大学"/>
  <tag k="name:zh-Hant" v="香港中文大學"/>
  <tag k="short_name" v="中大 CUHK"/>
  <tag k="type" v="multipolygon"/>
  <tag k="wikipedia" v="en:Chinese University of Hong Kong"/>
</relation>
```

Figure 3-3 Building attributes

### 3.1.3. Mercator Projection

Because the latitude and longitude information stored in nodes in the WGS84 reference system, whose standard is based on specular to represent the earth. If access the data directly and render into the plane, generated map will warp according to the longitude, magnifying in closer to equator regions and shrinking in around pole regions. Mercator projection provides a way to transform the specular coordinate into 2D plane coordinate.

### 3.1.4. Different types of region

According to analyzation on open source map data, CUHK map can be divided into multiple regions by land usage, such as water area, wood, building, restaurant, bus stop, etc. we have designed abundant and attractive game rules to collect props or occupy for more manor to their colleges in different kinds of regions.

In addition, except the main campus, buildings are divided into nine colleges, and each college has its own dormitory campus. To make the game equitable, because the region acreages have relatively large difference between colleges, for example, SHHO and SHAW college, only one or two buildings with similar area are not public for contending and always owned by its college.

### 3.1.5. Building

Frequently contending on buildings between different colleges is the one of attractive attributes in our game. How to enrich the information of each building according to raw data is our target. In currently development stage, building owns multiple attributes, like longitude/latitude, building usage type, height/levels, name with Chinese/English versions, and installations adding by players. In order to increasing user experience, the building occupied college is rendered with multiple colors for clear indication.

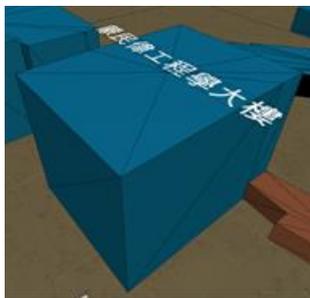


Figure 3-4 A building with default color

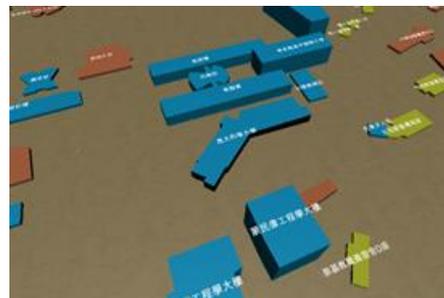


Figure 3-5 Buildings with different colors

For each building description in OSM file, its geometry attribute is formed by a closed way, with a series of nodes for lat/lon and the first is equal to last, which only provides multiple polygons in 2D with the coordinate of vertexes. In our research of mesh render principles in Unity, there is only basic triangle mesh rendering provided in pipeline. The mesh is formed by a series triangle, each of which contains the normal vector, index of 3 vertices and a list of vertex coordinates. According to the building height information and the radio of translation, each building 3d model can be divided into the roof rendered by triangulation on the polygons in way descriptions, and multiple side walls by each pair of neighbor nodes in the way.

When player touches the building in 3D virtual map, the building information surface, consists of 3d preview image and some basic information. The building preview image is acquired from a temporary camera, keeping a proper distance from the target building, which is only to capture the single building by pre-setting the culling mask on camera options, and then save in render texture and shown in the interface.

### **3.1.6. School Bus Route**

Currently for CUHK school bus, we can acquire for 4 route lines with the bus platforms along the way. When user is closed to the bus stations in real world, they can access the virtual representation objects for bus route, timetable, and next station information, which provides convenience for the tourist or freshmen who may be not familiar with CUHK route. With future database development, more school bus routes will be added into our game.

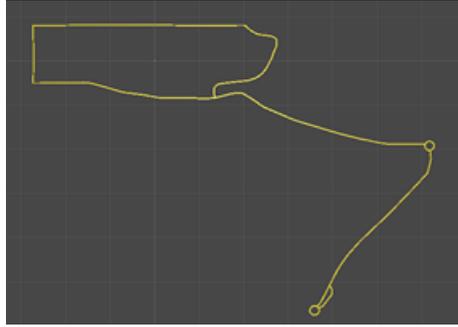


Figure 3-6 Route simulation

### 3.1.7. Road

Unlike to region description in OSM file, most the ways storing road information are open, that is, the series of nodes are like unclosed lines. The rendering pipeline has some difference from others, and because of lack for width of various road, each road is divided into multiple sub-rectangles, consists of two triangles, with the difference of adjacent nodes as length and preset value as width.



Figure 3-7 Road rendering

### 3.1.8. Others

There exist more other types of regions in the raw data and according to our game design, water and wood are what we have analyzed and rendered in current stage. The rendering pipeline is similar to the building roof render with triangulation for polygons.

## 3.2. Attack and Defense System

### 3.2.1. UI Design



Figure 3-8 Building information



Figure 3-9 Screenshot of attack mode

When players enter attack mode, the camera will open automatically which indicates the AR environment. At the top of the screen, some texts will be displayed which can prompt players what to do next. In case that players want to go back to map mode, they can click on the return button on the right corner of the screen and leave attack mode.



Figure 3-10 Attack UI



Figure 3-11 Defense UI

Firstly, players need to find a plane to place buildings. Once it's done, the 3D virtual building will be displayed in real-world environment and the weapon buttons will show up. Players may click on the buttons and use different weapons to attack the building. Players can also double click on the building of its installations to view their life point. And if the building's life point goes down to 0, a panel will show up which indicates that the player's college has occupied the building.



Figure3-12 Other players attack



Figure3-13 Other players defense

Moreover, as our game is a multi-player online game, players can also see other players around the building who is also currently attacking or defending this building. If other players are adding defense installations to the building, the player can also see the updates in his/her own scene. In addition, if other players are attacking the building, players can see arrows or bombs shot from other players.

### 3.2.2. Attack weapons

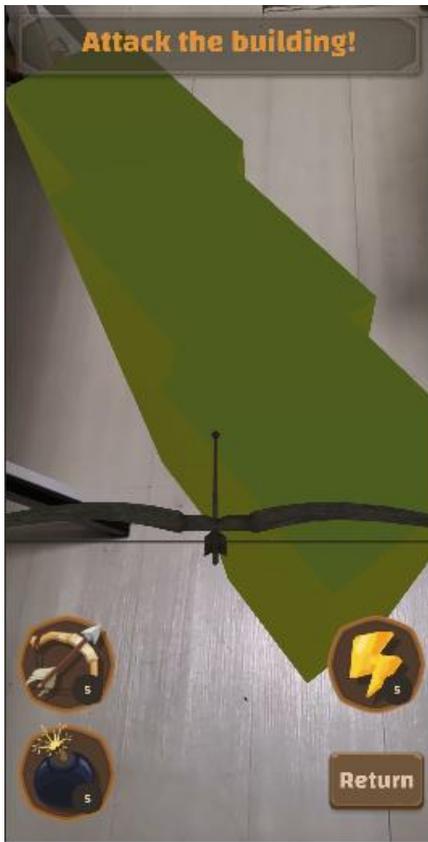


Figure3-14 Bow and Arrow



Figure 3-15 Catapult and bomb

We designed two types of weapons. One is bow which shoots arrows to buildings. The other one is catapult which shoots bombs to buildings. Arrow has damage of 10 but can only hit single object while bomb has damage of 5 and hits an area of objects. The price for an arrow is one wood and one fish and for a bomb is one wood and one mineral. The operating modes of different weapons are also different. To shoot an arrow, players need to touch on the bow and then move fingers to control the shooting direction. As players leave fingers from the screen, the bow will shoot the arrow at once. Shooting a bomb is relatively harder than shooting an arrow. Players need to slip on the screen vertically from down to top. The shooting speed and direction are decided by the slipping distance and speed.

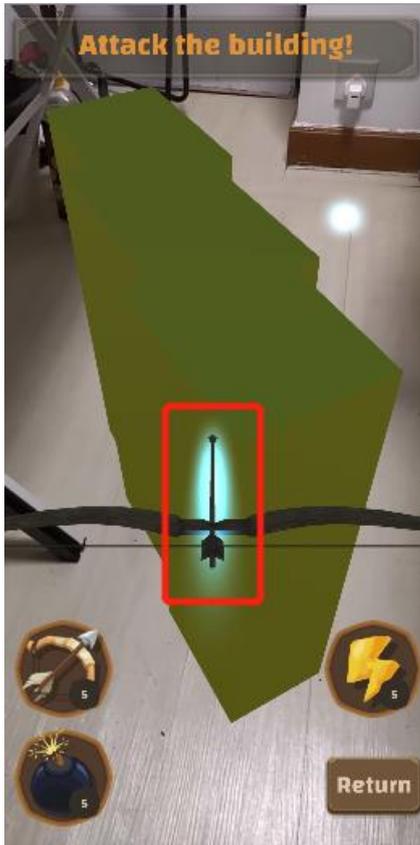


Figure 3-16 Intensified arrow



Figure 3-17 Intensified Bomb

Players can use power materials to intensify both two weapons. The intensified arrow has a higher damage of 20 and can penetrate objects which means it can take damage to all objects on its trajectory. The intensified bomb has a higher damage of 10 and a wider range of hitting area. Since collecting power materials is harder than other kinds of materials, players need to use them properly when attacking buildings.

### 3.2.3. Defense Installations



Figure 3-18 Defense shield



Figure 3-19 Defense wall

We have designed three types of defense installations. The shield can only be added to buildings and when it exists, the building will not be damaged. One shield can withstand 50 damages for the main building. To add a shield, players need to touch the main building first and then click on the add-shield button. Then a protective shield will be generated in the center of the main building and enclose it. The cost of one shield is one power material.

Another type of defense installation is wall which can be placed around the main building and withstand damages for it. When players click on the add-wall button, a wall will be generated above the main building. Players can change the wall's position and rotation by sliding on the screen or clicking on the rotate button respectively. Considering that if all players add walls to a building, it may be kind of

messy. So, we linked the height of the wall with its life point. Players can click on build button to make a wall higher which can better withstand damages. Finally, if players click on confirm button when adding walls, the updates will be recorded and send to server as well as all other players. On the contrast, if players don't satisfy with this adding operation, they can just click on cancel button and the new wall will be destroyed. The cost of a wall is five wood materials and five mineral materials.



Figure 3-20 Defense alarm

The last type is alarm. It is designed to inform players if the building is under attack. The alarm can be added to both buildings and walls. And when a building or walls are damaged by other players. There'll be alarm notifications sending to all players belonging to the occupying college. Alarms coexist with its attaching building or wall and will not be hit by any kinds of weapons. The cost of an alarm is one fish material, one wood material and one mineral material.

## 3.3. Collect System

### 3.3.1. UI Design



Figure 3-21 Entering collect mode



Figure 3-22 Finish collecting

When players enter collect mode, the camera will open automatically which indicates they are in AR environment. At the top of the screen, some texts will be displayed which can prompt players what to do next. In case that players want to go back to map mode, they can click on the return button on the right corner of the screen and leave collect mode. As players move their phones to find planes to place virtual objects, virtual planes will be generated above the real planes which make the environment more realistic. When players successfully collect materials, a panel will appear and show the type and the number of the material they got. Then they can click on return button and go back to map mode with their inventory being

updated.

### 3.3.2. Basic Materials

There're four kinds of materials and each of them has different way to collect. To encourage players to visit more regions in CUHK, all basic materials can only be gathered in special regions. In map mode, at the top right-hand corner of the screen, icons will flicker and indicate which type of materials that players can gather at this moment. Then they can click on it and enter collect mode.



Figure 3-23 Collect wood material



Figure 3-24 Collect water material

The first kind of material is wood material which can be gathered near green regions in the map. After entering collect mode, players need to find a plane and place virtual trees. The rule of collecting wood materials is that players need to cut the tree as much as possible in 30 seconds by sliding on the screen. Finally, the

number of materials players obtain relates to the number of cutting the tree.

The second type of material is water material and can be collected near water areas in CUHK. After entering collect mode, water surfaces will be generated as the background when players are finding planes. When people go fishing in real world, they must control and receive the rod properly. As a simple simulation, the gameplay of collect water materials is that players must control the slider stopping at a proper area by touch on screens. The number of materials players obtain is decided by the stopping position.



Figure 3-25 Collect mineral material



Figure 3-26 Collect power material

Another type of material is mineral material and can be collected near roads. The way of collecting mineral is relatively easier since this material is the most common one and we don't want players spend too much time on it. When players

move their devices to find planes, mineral objects will randomly come out. They only need to align minerals with the center of the screen and then click on it.

The last type of material is power material which is the rarest materials. It can only be collected near playgrounds. To collect it, players need to look around and find energy elements in the air and click on them. The time limitation is 30 seconds.

### 3.3.3. Inventory and Shop



Figure 3-27 Inventory panel



Figure 3-28 Shop panel

Players may view their bag information by clicking on the inventory button in map mode. The information keeps updating as players use materials or obtain

materials. To further know the detail description of an item, they can just click on the corresponding icon and view. Apart from basic materials, weapons and defense installations can be bought in shop menu by clicking on the shop button. The bigger icons are the weapons or defense installations, and each is followed by several small icons which are the costing materials. If players have enough materials to buy an item, the corresponding buy button will be interactable.

## **4. Research in AR**

### **4.1. Overview**

The word “augmented reality” was born and first used in 1960s (4), which builds a bridge between the virtual reality and the real environment. Combined with the “Virtuality Continuum” concept introduced in 1995 by Milgram and his team, AR becomes more understandable. They used this term to construct a “road” starting from a real environment, totally ending into virtual environment and passing through augmented reality and augmented virtuality. Therefore, augmented reality is the first stage in which virtual objects are superimposed on a real scene. The main difference between augmented reality and virtual reality is that, VR application tries to replace the environment with virtual objects but without any direct connection with surroundings, while AR application adds computer-generated digital information based on surrounding environment and shows the enhanced view of environment with virtual objects to users. AR development is still in its initial stages and with the advances of technology, more user-friendly AR applications will work on mobile devices and bring AR closer to the general public.

### **4.2. Mobile AR Development Framework**

#### **4.2.1. Overview**

We have do researches on the frameworks available to implement AR mobile apps on Android devices and an overview of this research is described below. The following description is captured from the characteristics and features of each framework which is out the most.

Framework	Last version found	2D recognition	3D recognition	Geo-location	Cloud Recognition	SLAM
ARCore	30 Oct,2019	✓	✓	✓	✓	✓
Vuforia	1 Mar, 2018	✓	✓	×	✓	✓
EasyAR	7 July, 2019	✓	✓	×	✓	✓
ARToolKit	--	✓	✓	×	×	×
Maxst	2 Dec, 2019	✓	✓	×	✓	✓

And in the following we would like to summarize the features of these popular AR platforms in alphabetical order as they appear in the table.

ARToolKit (5) is an open-source AR software library providing support the following functionalities: Single-camera or stereo-camera camera position/orientation tracking, tracking code that uses simple black squares, the ability to use any square marker patterns etc. It brings in OpenCV support, OpenGL ES2.X support, integration with GPS and compass, and automatic camera calibration utilities. ARToolKit uses computer vision algorithms to solve tracking from users' viewpoint, which is extremely fast to make camera pose estimation.

ARCore (6) is Google's proprietary AR SDK. It possesses three core capabilities to link reality and virtual world: Motion tracking, environmental understanding and light estimation.

EasyAR (7) is a free AR engine developed by Visionstar Information Technology (Shanghai). Except a number of basic features, such as SLAM, 3D tracking, and screen recording, it pushed out surface tracking in the newest version 3.0.0, a new kind of tracking that is calculated by detecting arbitrary surface feature points instead of costing extra time on finding plane and will enable faster surface bonding and pose tracking.

MAXST AR SDK (8) is a cross-platform AR engine, and provides instant tracker, visual SLAM, object, image, marker and QR code tracker, QR reader and cloud recognizer. In addition, MAXST combined all the trackers into AR Fusion Tracker in the newest released version and used ARCore to track environment. Unlike other AR engine, MAXST use octagonal bounding box to compensates for user tilt between the camera and scanning object in order to make the Visual SLAM more accurate.

Vuforia (9) is the AR SDK that enables app developers to quickly get start because of the complete and detailed API. And it owns the unique tracking technology, 3D model tracking, and provided the option to train model targets using Deep Neural Networks(DNNs) for recognition in recent version, and the result of training is used to not only identify whether the object is matched, but what angle the camera viewing it from (10).

In this semester, because of time limitation, we only concentrated on two of these technologies, Vuforia and ARCore.

#### **4.2.2. Vuforia Engine**

Vuforia Engine is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. Among many of the AR software development kits, Vuforia Engine is the most popular one with the fastest recognition speed and the most stable recognition effect. There is little detailed algorithm on Vuforia official website, our researches concentrate

more on the function and effect on recognition.

Vuforia Engine's recognition and tracking capabilities can be used on a variety of images and 3D objects, but only including model pre-existing in database or the sample geometry like cylinders and cubes.

### 4.2.3. Image Recognition

For images, before recognition, users need to upload the target image to Vuforia database, which will analyze rating of the image(from 0 to 5) by the amount of feature points. Here are two examples with different grades.



Figure 4-1 Images of feature points with rating of 0(left) and 5(right)

By a mount of feature points, Vuforia understands the unique elements the target images own, such as curves, edges and high-contrast spots, which can help distinguish whether it is easy to recognize. After analysis of target images, Vuforia engine operate the images captured by physical camera from each frame, search the feature points, and matched if majority of reference feature points are found. The next step is calculating the relative position of all feature points and combining

the virtual game object with relevant position information. Here is an example showing how matching the reference feature points between camera captured image and preset image.

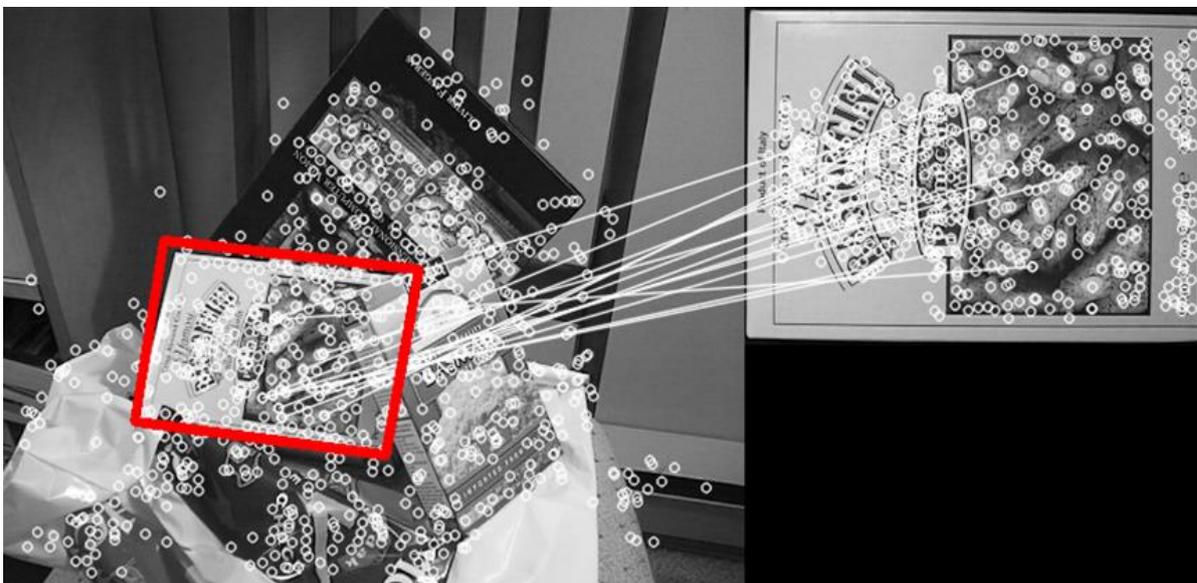


Figure 4-2 matching feature points between target image and camera capture image

#### 4.2.4. 3D object recognition

To make a 3D model target for a particular object with high accuracy effect, Vuforia has some limitation of the object, be geometrically rigid, present stable surface features without shiny surface and be fixed with respect to the environment it is in (9). First through Model Target Generator (MTG) provided by Vuforia engine, the 3D model data representing the object users want to track would generate into a Vuforia Database, and the input data only can be accepted with 3D CAD model file, or using the Vuforia scan of the object with specific angle and distance. Currently because of lack for the digital model for each building and the difficulty of scanning each building with global angle, we do not touch this part during this semester and may try to train our database via cloud-based deep learning based on the images captured by multiple players in our future plan.

#### **4.2.5. Advantages and disadvantages**

Vuforia is the most powerful tool with the good integration into Unity, and the API is enough detailed to facilitate and simplify the development process. But the recognition with high accuracy has too much limitations, for example, when we do the plane finder, only when having contrasting coloring, or some details, the plane can be recognized. In addition, any markers with striped element will not be recognized in any case. How to avoid these disadvantages is our next step.

#### **4.2.6. ARCore**

ARCore is Google's platform for building and bettering AR experiences, bringing information and content to life at the exact moment it's useful. Its functionality can be split into three main areas: Motion tracking, Environment understanding and light estimation (6). Light estimation allows the device to estimate the surrounding light information of environment and calculate the virtual objects texture to naturally blend into the scene, which may not be applied in our design. So, we focused more on the first two features.

#### **4.2.7. Motion tracking**

From the ARCore official API, motion tracking is implemented by a process called parallel ranging and mapping (or COM). ARCore detects visually distinct features (feature points like Vuforia) in captured camera images and uses the difference between these points to calculate their position changes and then combined with the inertial measurements of the device's IMU to estimate the pose of the camera relative to the surrounding world over time. IMU of mobile phone

depends on following hardware.

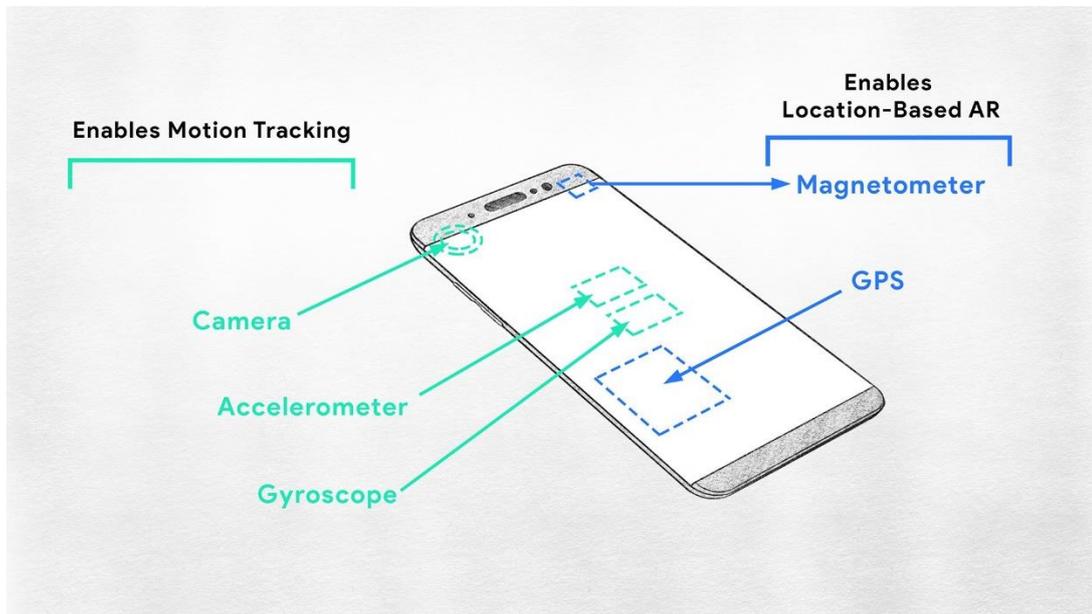


Figure 4-3 Hardware of mobile phone enhancing the motion tracking

- Accelerometer: measures proper acceleration which is felt relative to freefall by object itself.
- Gyroscope: measures and maintains orientation and angular velocity. It can calculate the rotation and ensures the virtual rotate respond correctly.
- Magnetometer: provides a simple orientation related to the Earth's magnetic field so that the virtual world map can auto-rotate depending on the physical orientation of device (11).

Only using IMU for navigation is that the tiny error will be accumulated because this system is continually integrating acceleration with respect to time to calculate velocity and position based on its last calculation without any static basis, that means, no matter how small the measurements error will be accumulated over time. This leads to “drift”, an ever-increasing difference between the pose in real world and in virtual world. Due to integration a constant error in acceleration results in a

linear error in velocity and a quadratic error growth in position and a constant error in attitude rate results in a quadratic error in velocity and a cubic error growth in position (12). s

Position tracking systems such as GPS can be used to continually correct drift errors.

#### 4.2.8. Environment understanding

ARCore enhanced the understanding of surrounding real world environment by detecting feature points and planes. Because one of the shortcoming of ARCore is that it doesn't provide any visual and detailed recognition process, the feature points or planes cannot shown either. Only a cartoon on the website may indicate the basic principle.



Figure 4-4 Process on how to find a plane

ARCore looks for clusters of feature points that appear to lie on common horizontal or vertical surfaces and by the position distribution from closer cluster, it can determine the plane position and edges of each plane. But similar to the Vuforia, because the flat surface without any texture cannot provide any feature points, ARCore also has no understanding on such plane.

### **4.2.9. Choose of tool**

In the first semester, because of lack for ARCore support devices and convenience development environment, Vuforia is our main research direction. However, as further research on the technology core in AR, the encapsulated functions provided by Vuforia Engine are not suitable for our development. In addition, for non-profit organization, Vuforia only provides limited database and online temporary memory, which is also a constraint in our application.

In this case, during our second semester research, we convert our application into ARCore Tool, which provides more underlying interfaces and functions. By studying on official API, raw data, such as point clouds, frames, anchor, pose and etc., can be accessed for further analysis.

## **4.3. Point Cloud for Classification**

### **4.3.1. Introduction**

Accessing a series of points coordinate from ARCore point cloud structure and depth values from camera device input, we explore deep learning architectures for 3D object classification. Because in previous work on 3D research, typical convolutional neural networks require highly regular input data formats, such as 3D voxels, mesh, to perform weight sharing and other optimizations on filter kernel. Since point clouds in our application are accessed from player camera input on real-time, the format and order are not regular, which is typically transformed into 3D voxel grids or collection of 2D view images in previous related work but the performance and elapsed time are unsatisfactory.

Learning from PointNets, a research paper published in 2017, we construct our neural network that directly takes point clouds as input, extracts the features for networking transformation, and out object classification result.

### 4.3.2. Related work

- **Point features**

As ARCore official on point cloud description, feature points are produced in the first frame and updated its confidence along with motion tracking and frame changing, then according to the limited memory of point cloud, points with the top confidence are updated into the collection structure. Because there exists no detail on confidence calculation disclosure from ARCore tool, in general, as frame frequently change frequently, points features should be encoded and designed for invariant to certain transformation to keep the pose robust while processing. However, ARCore only provides feature points extraction but no other feature analyzation.

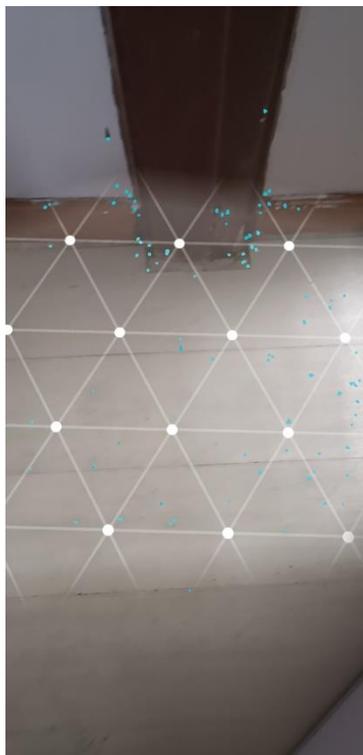
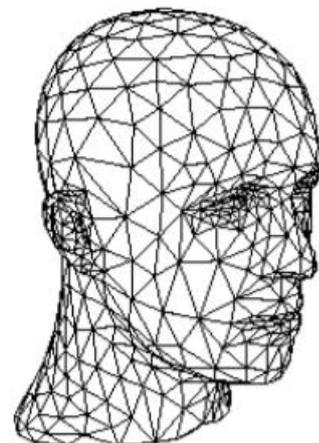
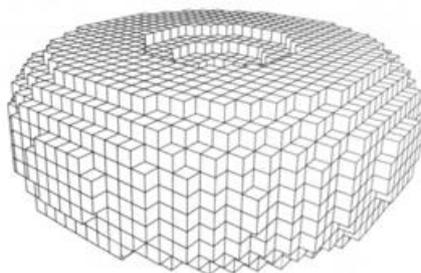


Figure 4-5 ARCore point cloud

- **Properties of point cloud**

There exist various kinds of structure for 3D object representation, such as point cloud, polygonal mesh, voxels, and other high-level structures. Each representation has enough expressive power to model the shape of geometric object but has different efficiency for different tasks. For point cloud, the unstructured set of 3D point samples, has more accurate architecture but non-robust transformation, that is, because different orders of points stored in point cloud may be represented the same 3D objects but perform different result after transformation. So that symmetric function design, performing to the same output, is one of the keys to our approach.

In addition, interaction among neighbor points is also a property of points cloud, that is, each point is not isolated but form meaningful local structure with neighboring points, like edges, smooth regions. Therefore, how to keep the local structure features into extract global feature is also in concern.



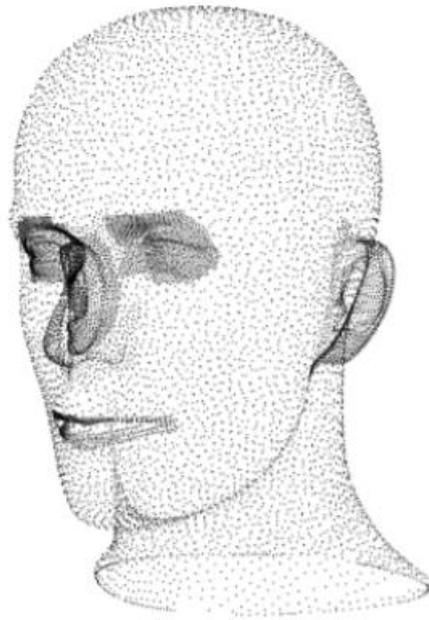


Figure 4-6 Structures of 3d object (voxel,polygon mesh and point cloud)

- **Deep learning on 3D data**

As 3D data has multiple popular representation as mentioned above, leading to various approach for learning. *Multi-View CNNs* (13) generate 2D images from multiple views of a 3D shape and apply CNN on images for shape analysis, but the feeding data require high quality and quantity of images. *Voxel-based CNNs* (14) model the 3D shape as a function sampled on voxels and define a 3D CNN over voxels for shape analysis, however, volumetric representation has great constraints on voxel resolution and the calculation cost relatively large computational memory.

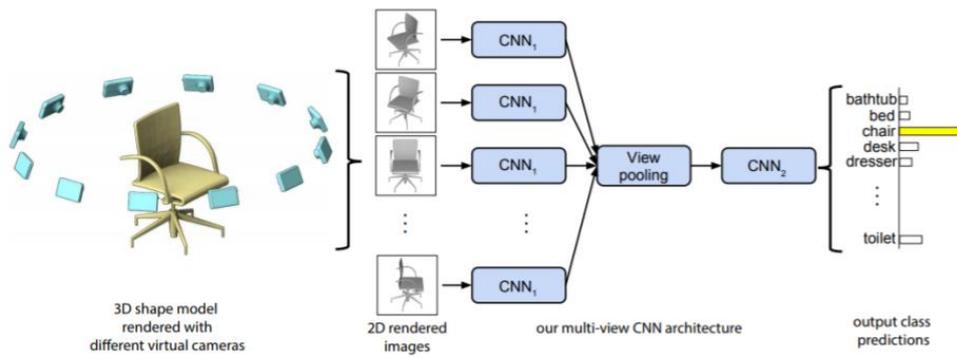


Figure 4-7 Multi-view CNN

### 4.3.3. Network structure

Our full network structure is visualized in figure below, where the whole network can be divided into two parts, encoder E and decoder D, with the key parameters annotated below each module. The encoding neural network E converts any points cloud in xyz-coordinate to a 128-dimension vector  $V$  as its global feature representation. Conversely, the decoding network D converts it back for reconstruction of points cloud for 3D object.

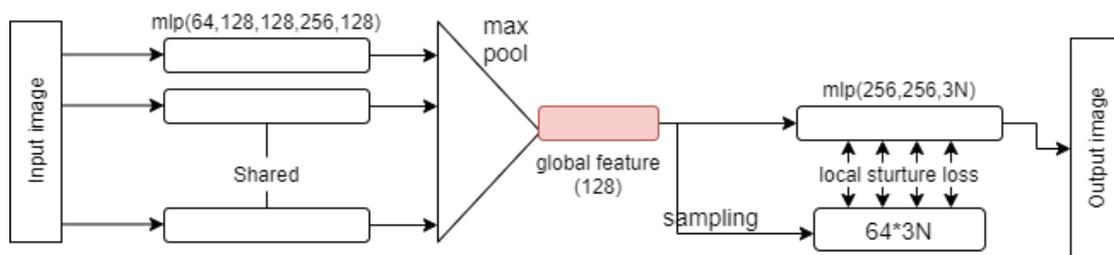


Figure 4-8 Encoder and Decoder

In order to make the model invariant to input permutation, here we apply max-pool function as the symmetric function for unordered input. That is, no matter the order of input data, applying the max function and the generated results are the same.

The idea is to approximate a general function defined on a set of points by applying a symmetric function on transformed elements in the set:

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n))$$

Where  $f: 2^{R^N} \rightarrow R$ ,  $h: R^N \rightarrow R^K$  and  $g: R^K \times \dots \times R^K \rightarrow R$  is a symmetric function.

Empirically, the basic module is very simple: after multi-layer perceptron network to approximate function  $h(\cdot)$  and composition of single variable function following with a max pool layer to generate as  $g(\cdot)$ . This is found to work well by experiments. According to the collection of  $h(\cdot)$  function, we can learn a number of  $f(\cdot)$  to capture different features extensively (15).

#### 4.3.4. Objective function

- **Invertibility Loss**

Given an input point set  $I$  and the reconstruct point cloud output  $O = D(E(I))$ , the invertibility loss  $L$  ensures that  $I$  and  $O$  are as similar as possible. Here we just simply utilize the CD distance to evaluate the difference, which form the main part of objective function.

- **Local structure Loss**

This sub-loss aims at preserving the local structure of the reconstructed point cloud. That is, if the region structure is locally smooth in input, it should keep smooth in the generated grayscale image. To measure the structure similarity, prediction layers of each 64 points local position and features (in 128-dimension) are added before the reconstruct multi-layer perception blocks. With the local features of each point subset, the prediction of local position of the (total number/64) points in each subset and finally transform them into the global relative position combined with the first prediction. This loss is calculated by the local CD to make the network take more concern on the local geometry.

#### 4.3.5. Classification algorithm

Here in object classification part, according to the pre-trained network, each

input point cloud has its own feature vector  $V$  acquired from encoder. Similarly, for each point cloud in query dataset, we can also obtain their feature codes. And by calculating the code difference (e.g. Euclidean distance) as the measurement of similarity, we could find the top-K point cloud representing the most similar object among query dataset. The pseudo-code is summarized below for reference.

---

**Input:** a well-trained network, a query point cloud and a searching database

---

1: compute the feature code of the query point cloud using a well-trained network  
2: **for** each point cloud in the database **do**  
3:   compute the feature code  
4:   calculate the code distance as the similarity  
5: **end for**  
6: sort and find the K most similar point clouds

---

**Output:** the K most similar point clouds for the given query point cloud

---

Figure 4-9 Pseudocode of classification algorithm

#### 4.3.6. Result and Discussion

- **Point cloud reconstruction**

We evaluate our method on point clouds of various genre and content, and in currently research stage, the global feature keeps high consistence between input and reconstructed output, however, the local structure remains obvious difference especially for the elaborate model, such as flower. Because of the reconstruction procedure is mainly based on the global feature. In addition, the local structure process simply samples the closest data into a group with certain amount, which may cause relatively large error in the model, with some points are dense and some are sparse. And for our future research direction, maybe sampling property subset, concentrating more on local structure, to improve our model performance.



Figure 4-10 Original input and reconstruct input



Figure 4-1 Original input and reconstruct input

- **3D object classification**

Our network learns global point cloud feature, as the algorithm mentioned before, which can be used for object classification. We evaluate our model on the ModelNet40 shape classification benchmark, which includes 12311 CAD models with 40 man-made object categories. Comparing with other mature 3D classification models, our model performance a little worse. But as the reference paper PointNet provided result, they have required pretty better result than other 3D model representation network, that is, our current direction may be correct and have a long way to improve.

	PointNet(2017) (15)	3D-CNN	WU's model (14)	Ours
Accuracy (average)	83.7	79.4	81.4	70.8

### **4.3.7. Conclusion**

In this work, we have proposed a novel deep neural network referenced from PointNet which performs directly consumes point cloud. In current stage, because of limited time, the network has not combined into whole application development and just as a sub-branch research on AR. In future continues study, our next research directions are to improve the model performance and realize into the game design.

# 5. Network

## 5.1. Overview

As a networking multiple-player game, how to synchronize the events happened in the world is necessary. In this project, because UNET, the Unity official packet for multiplayer game, has been deprecated, the protocol and the architecture of packets between server and multiple clients are designed by us. As the time limitation, our whole networking structure is from single server and multiple clients. From the client-side viewpoint, such as register personal information, share geometric position, attack, or defense, we have implemented the end-to-end process with various data packets. From the server-side viewpoint, centralizing the information and distributing synchronized game events are the major functions.

## 5.2. Server

Here is overall architecture of game server. It consists of connection listener, game logic processor, and database.

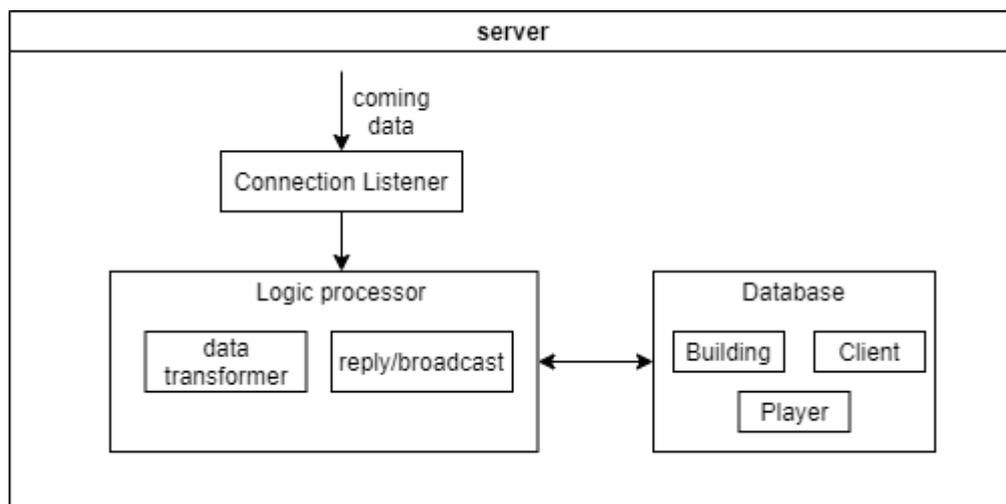


Figure 5-1 Overall architecture of game server

### **5.2.1. Connection Listener**

This is a predefined module that set up the server and listen to TCP incoming connection attempt and create the client module to handle remote host communication on specific port. This is the beginning of the whole system, the client requests are accepted and processed after the IP address rationality checking. After connection setting up, data between end-to-end is transferred by network stream, which transforms the 8-bit unsigned integer as the basic element. And the unprocessed raw data will proceed to main game server for processing.

### **5.2.2. Game Logic Processor**

Game server is mainly responsible for game logic processing. The network game has the huge world view background, the furious in-time faction confrontation as well as various equipment system. Acquiring raw data from socket listener, transform into different data type is the first step. In this project, we use integer, float, string as basic data type, which combines various kinds of data packet architecture. With predefined packet protocols, server process the packet and centralize data from multiple clients, and send back or broadcast the reaction through packing data and transforming into bytes with the existing pipeline.

### **5.2.3. Database**

In game server, the database defines a collection of fields and related data types. Notice that the data type and parameter are predefined in server and the access right is only owned by the server. The database now is implemented by dictionary with unique distributed key, and it will upgrade to junior data stored framework

like SQL dataset when the amount of data increasing in the future. Here are the details of the database models that we defined.

### **Building:**

Building information is mainly used for keeping the map consistence among all clients.

- **BuildingID:** The unique building ID distributed by server for identified during transforming.
- **Installation:** Dictionary of <ID, installation> with size equal to the no. of installations on each building, id keeps the peculiarity of installation, and each installation structure contains health, type, relative position and quaternion.
- **OccupiedCollege:** College ID to indicate the building belongs to which college.

### **Client:**

To store the network address of each connection including TCP socket and UDP endpoints.

- **Socket:** The TCP socket set whenever the client online and request for server data.
- **ReceiveBuffer:** Byte data array for storing temporary receiving to-processing raw data.
- **IPEndPoint:** Represents the network endpoint as an IP address and a port number of clients.
- **PlayerID:** The ID of current processing player.

### **Player:**

To store player information, initialized when each player sign into game at the

first time.

- **PlayerID:** The unique player ID distributed by server for identified during transforming.
- **Username:** String value storing the username uploaded by player
- **College:** Player subordinating college.
- **Position:** Updating player position in <float,float,float> data format.

#### 5.2.4. Detailed implement of functions

- **Player Register**

When the player log into the game, client should send a register packet to server to update its state. If the player is the first time playing, an extra packet with applying for a unique distributed id packet is also required.

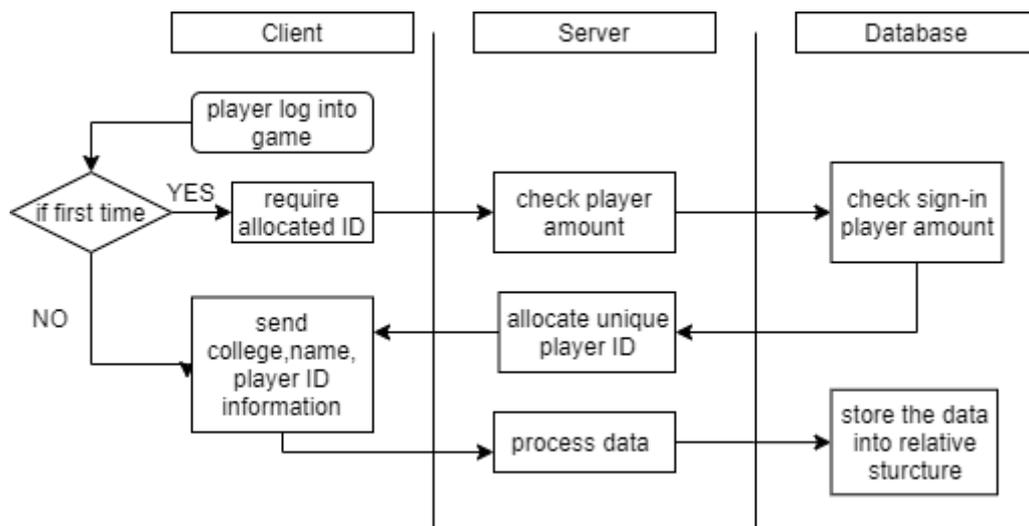


Figure 5-2 Process diagram of register

- **Attack/Defense Building**

When attack/defense operations on specific building, clients will send a building information require packet to access the building relative game data, including installations position, type, health and the amount. In addition, in order to let multiple player attack/defense on the same building simultaneously, switching into AR packet is also sent to server and broadcast to all other players who are operating the same building to update the game object in client AR scene.

- **Attack**

After 3d physical collision analyzation, the weapon crashed installation and damage are determined, therefore, a packet contains attacked installation or building id, weapon type, damage will be sent to server. In order to update the animation in AR scene for other player who are operating on the same building, server will trans pond the attack information to others doing operations concurrently, and the receive callback packet with the result of whether the attack target is destroyed is replied to the attacker.

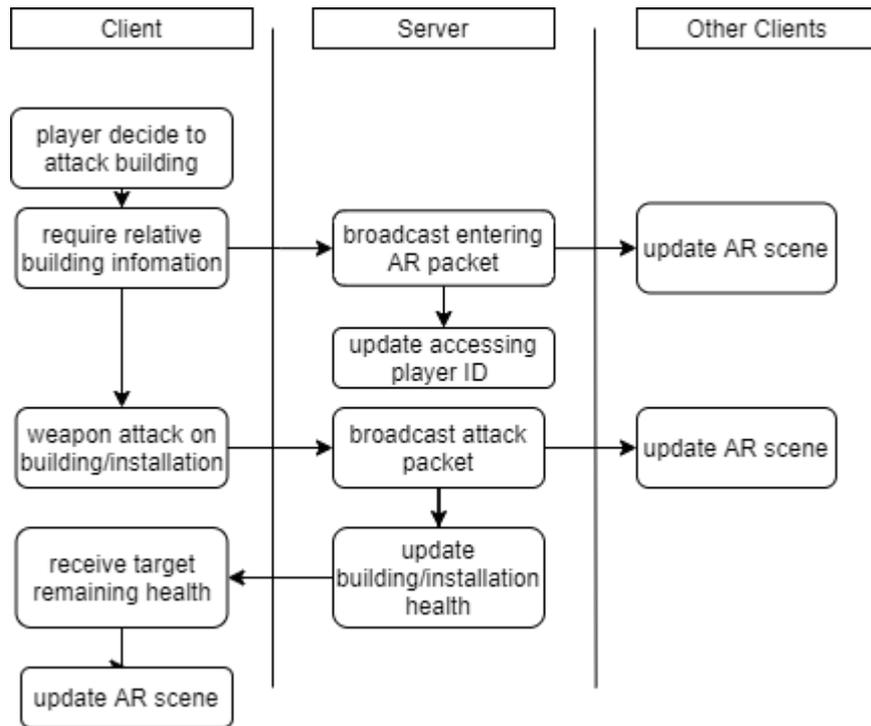


Figure 5-3 Process diagram of attack =

- **Equip Installation**

Similar to the attacking process, after the relative position acquired from AR scene calculation, clients will send the equipment packet including installation type, position, quaternion and initial health to server. Then server will send back the allocate unique installation ID and broadcast the equipment information to other player who are attacking or defending at the same time.

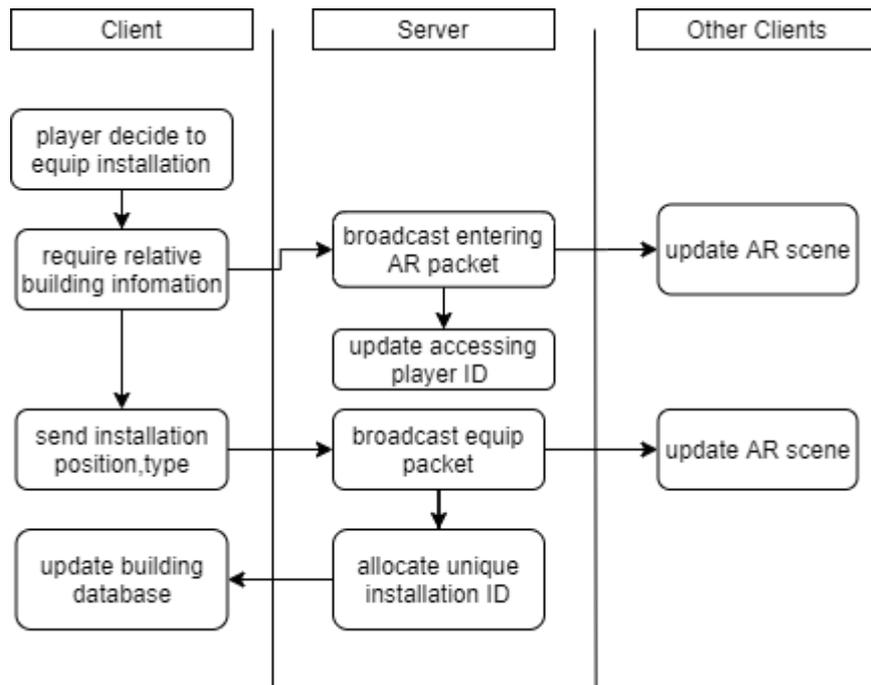


Figure 5-4 Process diagram of defense

- **Position update**

In the virtual 3d CUHK map, online players are shown with different colors to indicate their belonging college and real-time position. In this case, GPS position of each player need to be updated into server in regular intervals, currently set 10 seconds as a period. Whenever server receive the position from clients, it will broadcast the packet to all other online players to update their map.

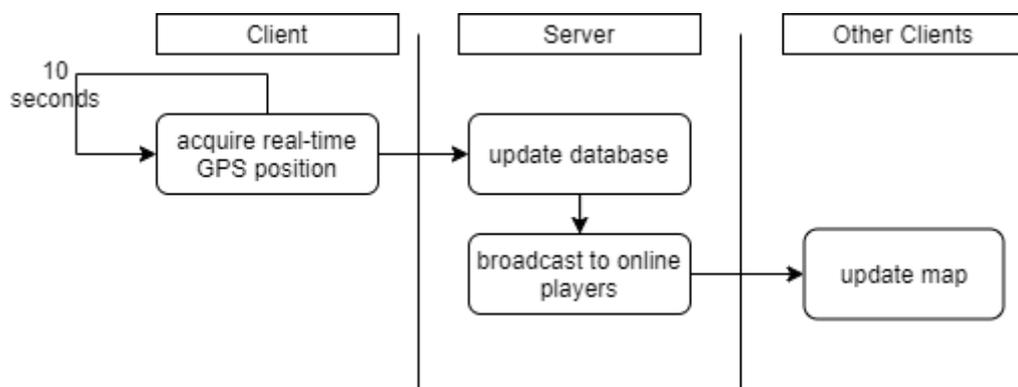


Figure 5-5 Flow chart of updating position

- **Disconnection**

When players close the application and decide to disconnection from the central server, clients will first send a disconnection server to inform the server. And after server received, the packet is also broadcasted to other online players for indication and its relative socket and memory are free after a short time.

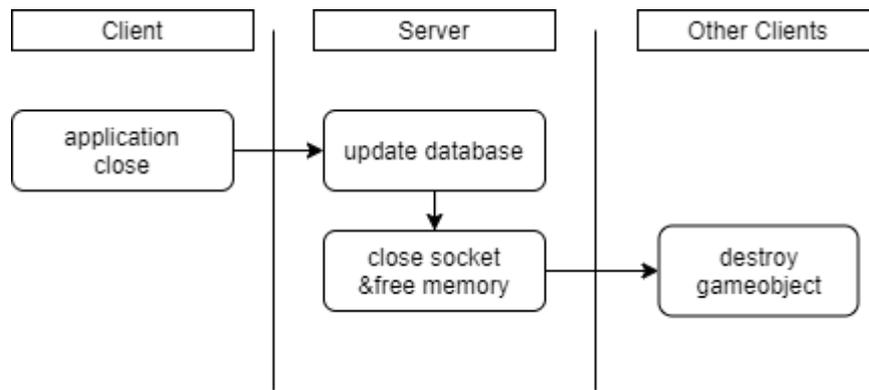


Figure 5-4 Process diagram of disconnection

# 6. Conclusion

## 6.1. Summary of second term

In general, the objectives of this semester have been achieved. First of all, although we knew that it's very difficult to develop online games, we still designed and implemented our game as an online game and developed network framework all by ourselves. Moreover, we integrated the network module with our game and fixed many of data transmission bugs as well as synchronization issues.

Secondly, to make our game more complete and more fun to play, we've designed four interesting ways to collect different kinds of materials in AR scenes. And we implemented two kinds of weapons and three types of defense installations. We came up with all these designs by thinking from the perspective of players. We considered a lot about playability, players' level of acceptance as well as operation difficulty for players.

Moreover, we also improved the artistic of our game a lot. As for the main map, in order to provide players with more reality experience, we restored the campus map with buildings, facilities, roads, and some natural region distributions. In addition, to unify the overall style of the game, we updated the whole user interface and replaced most of the resource materials.

To summary, we have developed an AR, multi-player online game with the CUHK campus as the map, CUHK colleges as the teams targeting for CUHK students and stuffs. We also added some special and interesting functionalities into the game to make it useful in some other circumstances. We hope, in the future, if we could release our game to public, people will love it and have fun with it.

## 7. Division of labor

This project was completed in two terms. In the first term, we designed the game and developed a framework of it and made some attempts in AR area. In this term, we devised the framework and added some new components including network as well as AR implementation. As our goal is to develop a complete game, the following table is a summary of our main accomplishments of each of us to the project in design, implementation and research perspectives.

Game components	Design	Implementation	Research
Location-based Map	Hardy & Daniel	Hardy	Hardy
Collection	Hardy & Daniel	Hardy & Daniel	
Attack & Defense	Hardy & Daniel	Daniel	
Network	Hardy	Hardy	
User interface	Daniel	Daniel	
Augmented Reality	Hardy & Daniel	Hardy & Daniel	Hardy & Daniel

Contribution to the report can be summarized as follows,

Index	Title	Person in charge			
		Daniel		Hardy	
0	Abstract & Acknowledgements	X	2		
1	Introduction	X	2	X	1
2	Design	X	8	X	1
3	Implementation	X	12	X	7
4	Resear in AR	X	3	X	14
5	Network			X	8
6	Conclusion	x	1		
7	Division of labor	-	-	-	-
8	Bibliography			X	
Total page count		28		31	

# Bibliography

1. **Negahban, Salehan &**. Social networking on smartphones: When mobile phones become addictive. 2013.
2. *Augmented reality: Research agenda for studying the impact of its media characteristics on consumer behaviour*. **Javornik**. 2016, Journal of Retailing and Consumer Services, pp. 252-261.
3. **B.Lovelace**. 'Pokemon Go' now the biggest mobile game in US history. *CNBC*. [Online] 2016. [Cited: Dec 6, 2019.] <http://www.cnbc.com/2016/07/13/pokemon-go-now-the-biggest-mobile-game-in-us-history.html>.
4. **Sutherland, I.E.** The ultimate display. In: proceedings of the IFIP Congress. 1965, pp. 506-508.
5. *ARToolKit*. [Online] [Cited: Dec 6, 2019.] <https://www.hitl.washington.edu/artoolkit>.
6. *ARCore*. [Online] [Cited: Dec 6, 2019.] <http://developers.google.com/ar>.
7. *EasyAR*. [Online] [Cited: Dec 6, 2019.] <http://www.easyar.com/>.
8. *MAXST AR SDK*. [Online] [Cited: Dec 6, 2019.] <http://maxst.com/#/en/arsdk>.
9. *Vuforia*. [Online] [Cited: Dec 6, 2019.] <http://www.vuforia.com/>.
10. Martin, Edward. Automotive advantage: Vuforia Model Targets. *Unity Blog*. [Online] Feb 1, 2019. [Cited: Dec 3, 2019.] <https://blogs.unity3d.com/2019/02/01/automotive-advantage-vuforia-model-targets/>.
11. Chopra, Shivang. Introduction to Motion Tracking in ARCore. *Medium*. [Online] Jul 23, 2018. [Cited: Dec 5, 2019.]
12. "Robot Navigator Guides Jet Pilots". *Popular Mechanics*. p.87, May 1954.