Virtual Reality Game Development using Unity and Leap Motion

VR Game Development using Unity and Leap Motion

FYP Final Report

by
LEE Chun On, SIN Wing Lam, WONG Chun Yin
(DES2)

Advised by
Dr. Desmond Tsoi

Submitted in partial fulfillment
of the requirements for COMP 4981
in the
Department of Computer Science
The Hong Kong University of Science and Technology
2016-2017

Date of submission: 20th April, 2017

Abstract

Virtual Reality (VR) game has greatly impacted the game industry recently. It greatly improved the sense of reality and excitement of games. However, most of the VR games are controlled only by head motions and played with single player mode. Although some equipment, such as Oculus Rift and HTC Vive, are able to control using handheld devices, they are too expensive for many people. These factors limit the popularity of VR games.

This project aims to introduce a new control method to VR games and play in a network-connected environment. We developed a multiplayer VR game, Treasure Island, that is played with Leap Motion, an external hardware that detects hand gestures. With the assist of Leap Motion, players are able to control the game with bare hands and is cheaper as compared to that of Oculus Rift and HTC Vive.

TABLE OF CONTENT

1. INTRODUCTION

1.1 Overview

Virtual reality (VR) game began with Nintendo announcing its gaming console, The Nintendo Virtual Boy, back in 1995 [1]. Although it was a failure, it marked the start of VR game. After years of development, "virtual reality" becomes one of the most famous words in Game Development Conference 2016 (GDC2016) [2]. Oculus Rift, Sony PlayStation VR and HTC Vive are competing to be the most influential VR gadget while their games, like "Lucky's Tale", "Farpoint" and "The Lab", respectively fascinate players a lot.

Although VR games have tremendous advancements since 1995, there are features that can be improved. First, most of the VR games are designed for playing individually, that players cannot interact with other players spontaneously. Second, handheld hardware is required to simulate tools or hands, and equipment are bulky and expensive. For example, HTC Vive includes a VR headset, a pair of handheld controllers and a pair of motion sensors, users are required to set up these set of equipment and holding the handheld controllers during the game. It is inconvenient for playing; also the high price and tedious installation can discourage players from buying.

This brings out the idea of our final year project, building a networked VR game that recognize player's hand-gesture without the need of any handheld hardware. This project aims to make VR game more realistic, interactive, and able to connect different players. Also, it aims to reduce the cost and space required for playing VR game, making the game more accessible to players.

1.2 Objectives

Project Goal

The goal of our project is to build a First Person Shooting (FPS) game with multiplayer option, using Leap Motion as a controller. For VR game, what excites the players most is the sense of distance provided by VR

technology, which can hardly be achieved by a normal monitor. To experience this, one of the best game genres is shooting. In shooting game, there will be high-speed objects moving around and they are mostly aligned with player's eye direction, which maximize the change of distance from player's point of view.

With this goal, this project will focus on following objectives.

1. To build a standalone First Person Shooting game running on PC
2. To allow players to play under VR environment
3. To enable players to perform in-game control without handheld hardware
4. To allow two players play together in the same instance via Internet

To achieve the first objective, choosing a suitable game engine is the crucial and essential step. It serves as a toolkit for us to build games. This project will be developed with Unity game engine. Unity is a cross-platform game engine that supports game development on both PC and mobile.

To achieve the second task, VR head-mount is carefully chosen to ensure it can fit our designs, fulfill our requirements and provide a comfortable VR gaming experience to players. The details of choosing a suitable VR head-mount are in section 2.2.3 (Implementation, Hardware design).

To accomplish the third goal, Leap Motion is used to replace handheld hardware. Leap Motion is a small device connected with USB that detects user's hand gesture. With Leap Motion, player's hand gesture can be recognized and be rendered in computer. This project utilizes Leap Motion's gesture recognition to perform in-game control.

For the fourth objective, Unity has a built-in networking system as well as some well-developed network framework from external parties. This project adopts Photon Unity Networking (PUN) framework for multiplayer feature. PUN is the most popular network framework in Unity asset store with a general 5/5 rating from over a thousand of developers [3]. It provides well-developed toolset for network-related development.

Technical Challenge

There are mainly two challenges we have to face, display connection of mobile and gesture restriction on Leap Motion.

First is the VR display. Due to the demand of computational power on 3D model rendering and the lack of Android or iOS support from Leap Motion, building a standalone VR game on mobile is not a preferable solution for this project. To cater demand of computational power and Leap Motion support, the game will be built and run on computer, while mobile is connected as a VR display. VR image has to be exported to mobile phone and gyroscope data needs to be transmitted back to the computer. Most of the related tools available in the market only offer image mirroring, where screen of the computer is mirrored to mobile's screen. Data transfer is one-sided and cannot cater our need of head motion detection. Moreover, the network performance and resolution are not satisfying as data are transmitted via WiFi connection. Our solution is to use Trinus VR. Trinus VR is a project aims to transform players' mobile into a VR head-mount. Although its developer community is small and subject to frequent changes because it is at its early release, Trinus VR is still the best solution to solve the technical challenge we faced when connecting VR head-mount.

The second problem is the restriction from Leap Motion. Leap Motion perform the best when user's palm is perpendicular to IR camera as it can easily outline the shape of human hand. However, when user's palm aligns with IR camera, it can barely recognize user's hand and more likely to treat that as a rod-like random object and discard it. Moreover, it does not have a horizontal perspective of user's hands. Therefore, when user's hands overlap, it cannot perform a proper recognition and most likely see that as one hand. To deal with this restriction, our design on game control has to filter certain types of hand gesture in order to minimize the impact.

1.3 Literature Survey

In order to build a VR game with Leap Motion as the control, we will first examine the trend of development of VR games, VR head-mount and Leap Motion. We will then discuss a few projects with combination of VR and Leap Motion, which are closely related to our project.

1.3.1 VR Game trend

Virtual Reality is a growing trend in gaming industry with increasing investment from many companies of a total of more than US$8.8B since 2012 [4]. It indicates that this emerging technology worth investigation and development.

There are many VR games in the market. As shown in Fig. 1, dominating game genres are First Person Shooter (FPS), Simulation and Role Play Game, while the target players are mostly 15-35 years old [5]. Apart from entertainment, VR can also help in rehabilitation for patients who need cognitive and physical exercises [6]. A great example is a Kinected-based VR exercise game for stroke survivors called Durian-Runtah [7].

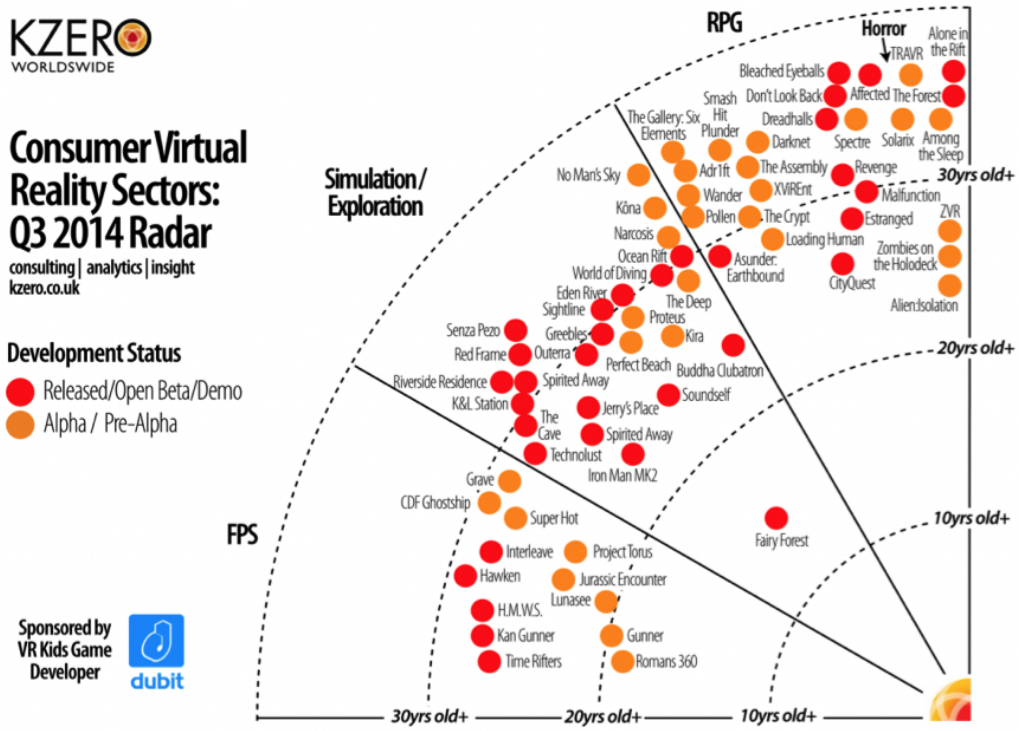


Fig. 1, Consumer Virtual Reality Sectors Q3 2014 Radar

As more than 75% of mobile game users in Hong Kong age between 16-44 in 2016 [8], FPS and Simulation/Exploration games are more suitable for our project. Meanwhile, FPS can provide more excitements in multiplayer mode, so a multiplayer FPS VR game is our project's target.

For in-game control, most of the existing mobile VR games are controlled by head movements. For example, the game "Lost In The Kismet" [9] is an escape game and the players can control objects and trigger conversation by staring at the object or person; another example is the game "VR Bike" [10], a Bike racing game, the motorcycle direction is controlled by the head directions of players.

There are also handheld or wearable game controllers available in the market, which allow more precise control from hand gesture. The game "Pillow's Willow" is created by Manus VR, which is a RPG game that players can control objects by wearing Manus VR gloves and moving their own hands [11]. The majorities of the existing VR games simply allow users to control with only head motion or handheld devices are required, while in our project, leap motion will be used to allow players to control more comfortably with their bare hands.

1.3.2 VR Head-mount

Market of VR is currently in a phase of intense competition. Since VR trend has just came out for 2 to 3 years, there is no dominating VR head-mount producer. Lots of technology companies seize this opportunity, and start producing their own VR head-mount. The following head-mounts are some of the popular products available in the market.

Oculus Rift is one of the biggest competitors in the market. It started as a Kickstarter campaign in 2012 and successfully raised over 2 million USD within one month [12]. It started shipping on 28th March, 2016 pricing at $599 USD. Besides the head-mount, Oculus also provides a device named Oculus Touch, which is a handheld controller, allowing button pressing and hand location tracking. Following Oculus Rift, another high-end VR head-mount HTC Vive is released on 5th April 2016, It is produced by famous smartphone company HTC, pricing at $799 USD. Similar to Oculus

Rift, it has a high hardware specification and a handheld controller for control. One extra feature is that Vive has room-tracking function [13]. Two infrared cameras are placed in the corners of a room and allow tracking user's position within the room. This allows user move around and movement is simulated in VR.

Oculus Rift and HTC Vive are both well-developed devices. However, the prices are too high for people who have no experience before and would like to have a try on VR. Therefore, this project adopt head-mount that uses smartphone as video display. After comparing multiple head-mounts including Samsung VR and Google Cardboard. Our final choice is a head-mount named VR Box, which supports smartphone up to 5.8 inches and can fit glasses.

1.3.3 Leap Motion and VR

Leap Motion [14] is a small USB device detecting user hand gesture with Infra-red (IR) cameras, emitters, and receivers. Hand gesture data are retrieved without requiring any physical contacts or touches. With the IR data, Leap Motion analyze and simulate user's hand model in computer with precision up to movements of every finger.

To combine VR with Leap Motion, there are developments going on from Leap Motion the company itself. It is launching its own project named Orion on providing support to VR head-mount. However, project orion only support Oculus Rift and HTC Vive which are some high-end VR head-mounts, and currently in beta stage. Because of that, this is not applicable in this project.

An approach on putting VR and Leap Motion together is to centralize data from VR gyroscope and Leap Motion to one mobile application for processing. As mentioned before, this would put too much burden if the application runs on smartphone. Besides, Leap motion SDK development for Android is currently in private alpha and no sign of development of SDK for iOS [15]. Building a standalone VR game on mobile has high technical difficulties and low payoff. Therefore, it is out of consideration.

This project adopts the approach on data centralization on PC. Leap Motion can be easily configured on PC as being one of the first platforms

to publish. Also, the Leap Motion SDK is sophisticated on Unity. For VR display, Trinus VR is used. Trinus allows smartphone to function like a VR head-mount with head motion tracking [16]. Leap Motion on PC with Trinus VR will meet the needs for this project.

1.3.4 Other Project with VR and Leap Motion

In August 2016, Leap Motion Inc. organized the World's First VR Dance Competition [17], reference to Fig. 2. It invited VR developers from all over the world to integrate Leap Motion into VR and have a dance party. This promotes Leap Motion Orion project and proves that there are potentials for Leap Motion itself to be a component in VR game.

Fig. 2, Screenshot of the World's First VR Dance Competition

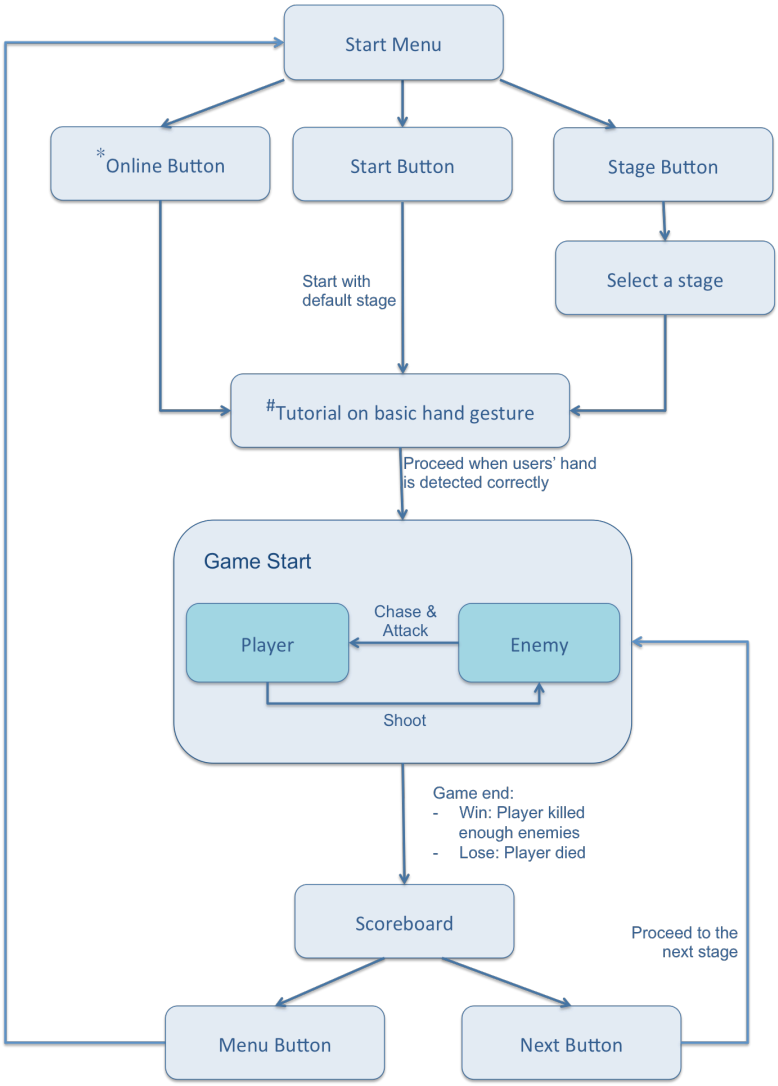
Most of the VR applications are only available to control by head motion or with other handheld devices, which reduces the sense of reality. Therefore, in our project, Leap Motion will be used to allow controlling with bare hands. The existing projects of VR with Leap Motion require relatively expensive equipment. Therefore, we will develop the game which is targeted to play with cheaper device like VR Box.

2. METHODOLOGY

2.1 Design

2.1.1 Game flow

The basic game flow is illustrated in Fig. 3, and the detailed implementation will be discussed in Section 2.2.1.



* Player can toggle between Online(multiplayer) and Offline(single player)
# Tutorial will not be shown again if player return to menu after player a stage

Fig. 3, Simple illustration of game flow

2.1.2 Model Design

The appearance of the game decides whether it attracts players, so we have designed scenes with different purpose as shown below:

i. Functional Scene

a. Menu Scene:

Fig. 4, Menu scene

b. Stage Selection Scene:

Fig. 5, Stage selection scene

c. Tutorial Scene:

Fig. 6, Tutorial scene

ii. Game Scene

a. Different stages of game:

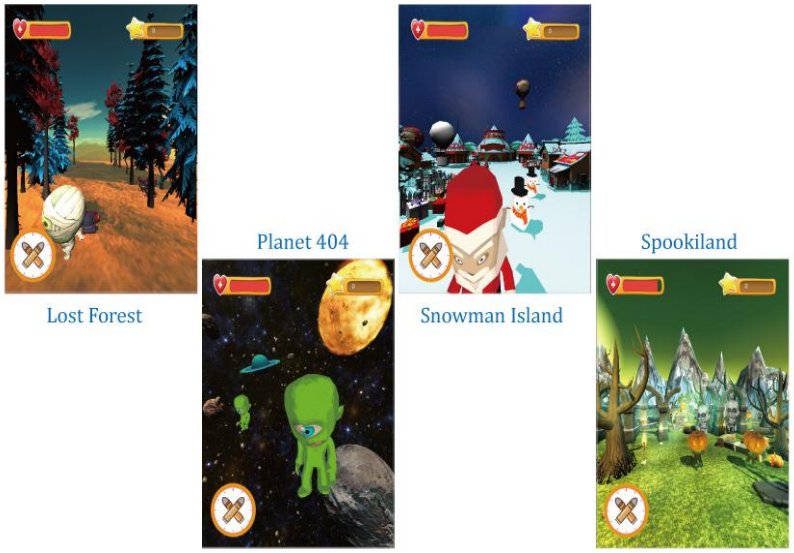
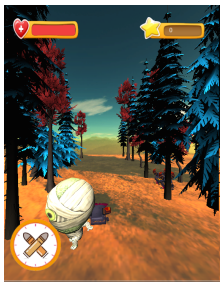
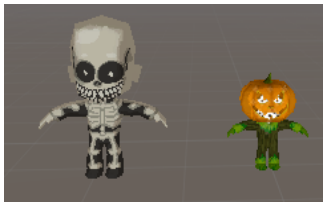


Fig. 7, Different stages of game

b. Table of enemies and weapons for corresponding scene

Scene	Enemy	Boss	Weapon
 Lost Forest	 Death Knight and Mummy	 Wood Monster	 Bullets, Knife and Lightning Shot
 Snowman Island	 Snowman and Santa Claus	 Bad Wizard	 Bullets, Fire Burn and Ice Wave
 Planet 404	 Alien and Astronaut	 Robot	 Bullets, Lightning and Fire Rock
 Spookiland	 Monster with skeletons and Pumpkin man	 Scary Monster	 Bullets, Dark Missile and Ice Wave

c. Score scene

Fig. 8, Score Scene after playing “Lost Forest”

2.1.3 Hardware Design

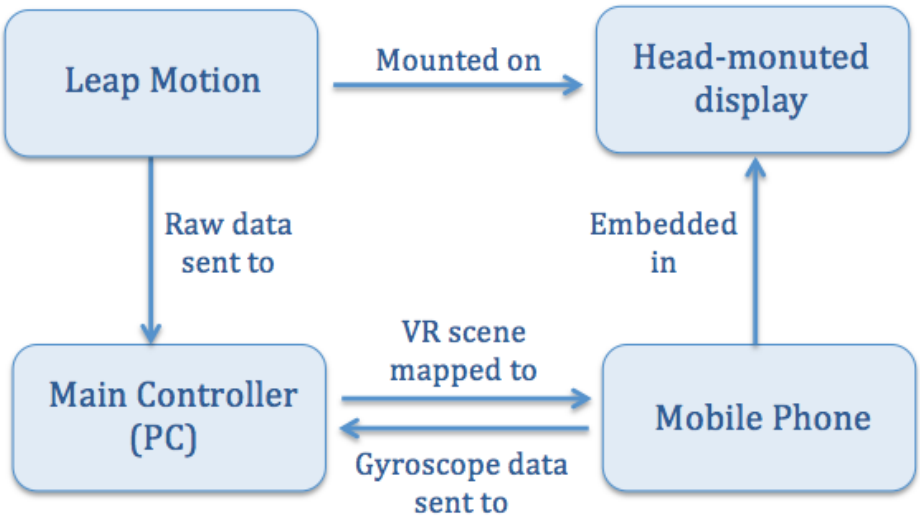


Fig. 9, Hardware components of the system

Fig. 9 shows the hardware components of the system and the relationships between different components. Below summarizes the relationships.

1. Leap Motion is mounted in front of the VR head-mounted display to capture hand motions.
2. Raw data from Leap Motion are sent to the connected personal computer where the game (Main Controller) is running.
3. VR split screen is rendered on personal computer and mapped to mobile phone through USB cable. Gyroscope data from mobile is sent back to Main Controller for simulating head motion.
4. Mobile phone is embedded into VR head-mounted display for the user to view and play.

2.1.4 Software Architecture

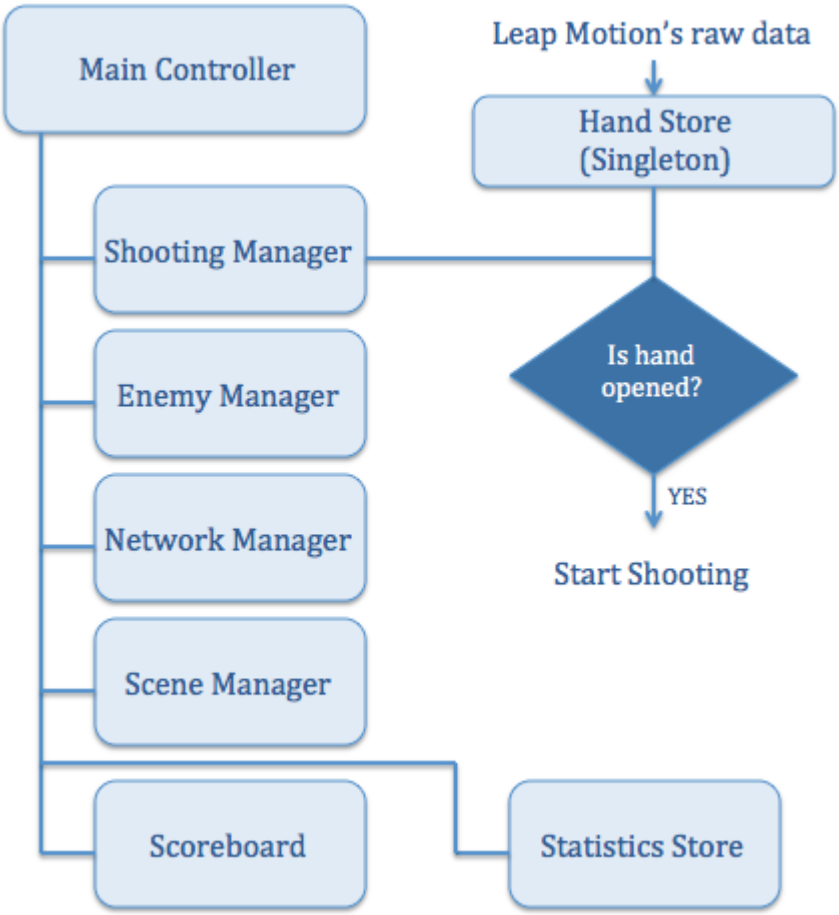


Fig. 10, Software architecture of the system

According to Fig. 10, the software architecture of the project is divided into 6 major components.

1. Hand Store
2. Shooting Manager
3. Enemy Manager
4. Network Manager
5. Scene Manager
6. Scoreboard and Statistics Store

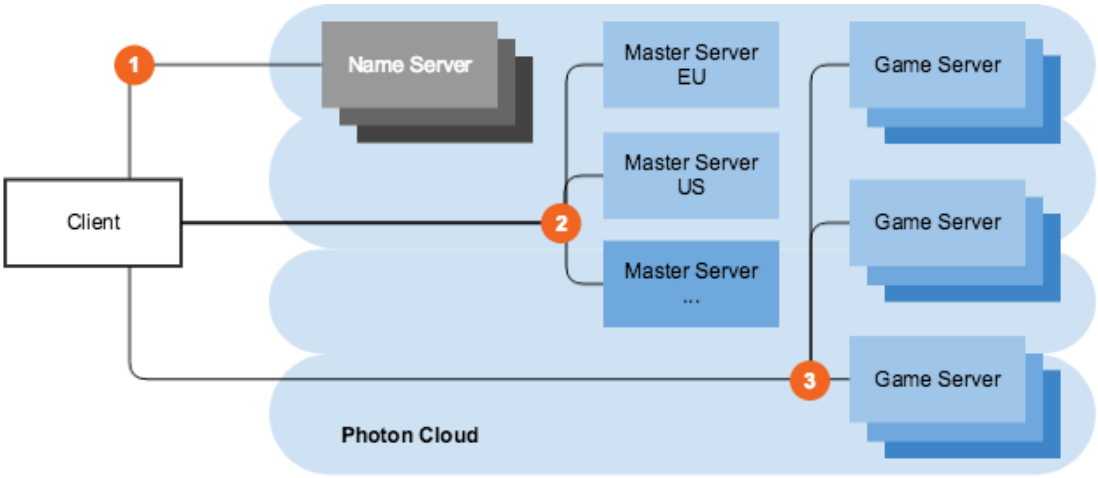
Hand Store (1) is responsible for receiving and organizing Leap Motion's raw data and then followed by vector, matrix interpretation to recognize different hand gestures.

Shooting Manager (2) collaborates with Hand Store (1) to give response to target hand gestures. For example, ShootingManager queries through HandStore's APIs to know whether the hand is opened. If yes, start shooting.

Enemy Manager (3) is required in every scenes to spawn enemies and manage spawning parameters, such as, monster types and spawning speed. Also, enemies are stage-specific, for example, Space Scene is having alien as enemy and Christmas Scene is having snowman as enemy.

Network Manager (4) is responsible for multiplayers mode. It ensures players are able connect to the internet and play together in a smooth, seamless environment with good user experience.

Regarding the network connection, our group is using Photon



Unity Networking (PUN) module.

Fig. 11, Structure of PhotonNetworking

PUN abstracts out details like IP address, socket, etc. It reserves resources in a private server based on client application ID that registered from Photon dashboard. The structure of PUN is shown in Fig. 11. When the network module in game client launches, PUN first connects to Name Server to verify if the application ID is valid. After that, PUN connects to Master Server which manages the

game server assignment. Lastly, it connects to Game Server and start creating or joining room for actual gameplay. Free plan of PUN provides only 20 concurrent users (CCU). When the project scales, we have to purchase the PUN service.

Scene Manager (5) is used to control all function related to transition of scene. It manages the flow from menu scene, through tutorial to the game scenes. It also defines the sequence of difficulties of different game scenes.

Sceneboard (6) is utility for recording and updating score for each player. It is also capable of updating components that has registered to scoreboard. Similar to scoreboard, statistic store is used to store game data such as number of enemy killed, player remaining health when game end. Score and statistics information will be shown after every stage.

2.2 Implementation

2.2.1 Game Flow

The game starts with a start menu, which player can select "Start" or "Stage". Player can either choose a stage from the map or start with the default stage (i.e. Lost Forest).

Then, a simple tutorial will be shown to teach players how to control with hand gesture, for example, shooting bullets with an open-hand gesture and changing bullet type with a shaking close-hand gesture. This tutorial will be shown once only during each time of game, therefore, when players finish a stage, go back to menu and start again, no tutorial will be shown.

The game starts by bringing the user into the game and starting to spawn enemies. Enemies will chase and attack the player; while player needs to shoot bullets with an open hand to attack the enemies.

The game ends when player win or lose. Player wins when the boss is killed; or loses when player's health hit zero. After the game ended, a scoreboard with gaming statistics will be shown and player can choose "Next" to proceed to the next stage or "Menu" to back to main menu. The stages order is "Lost Forest", "Snowman Island", "Planet 404", "Spookiland". In multi-player mode, player will be set to spectator mode if the other player is still alive. In spectator mode, player will be teleported to a high ground and no longer be able to shoot bullet.

To toggle multiplayer mode, player can use the "online" button in start menu. By looking at the "online" button for a few seconds, multiplayer mode will be turned off and "online" button will turn to "offline" button.

2.2.2 Model Design

i. Scenes

The scenes in the game can be classified into two categories, functional scene and game scene.

For functional scene (as mentioned in Section 2.1.2), the intuitiveness is crucial that players should know what they are going to do when seeing the scene for the first time. Therefore, they are very simple and straightforward. For example, there is one line of instruction will be displayed at a time in Tutorial Scene (as shown in Fig. 12).

Fig.12, Tutorial Scene with features indicated

For game scene (as mentioned in Section 2.1.2), they are the essential elements of the game, and we have implemented different themes in order to make the game more appealing. Apart from the decoration, each scene will have different background music, enemy and available weapon.

We have implemented most of the scenes. The map inside Stage Selection Scene will be updated with all the stages shown in it. Also, some new game scenes may be added into the game.

ii. Audio

Different background music was embedded into each game scene to fit the theme.

For example, a lively and happy music is chosen as the background music of Lost Forest (Link). A mysteric music (Link) with a slow tempo is used in Planet 404 to illustrate a space environment. A lively music with sleigh bells ringing (Link) as the background is used to improve the holiday feeling of Snowman Island. Finally, a spooky music that have a lower pitch background and some high pitch melody in slower speed (Link) to enhance the scariness in Spookiland.

Apart from background music, there are sound effect when enemies being hit, player get the present, and player finish a stage. Different types of sound effects are used to match the theme and action, such as metallic sound when hitting astronaut and robot enemies.

As the music are downloaded from different sources, the volume of them varies and we have standardized it to avoid manually adjustment during the gameplay.

iii. Enemy and Animation

Enemies are chosen based on the decoration of the scene. Therefore, each scene will have their own enemies to fit the themes. For example, in Spookiland (as shown in Fig. 13), the environment is scary so a pumpkin man and a skeleton on the right will be the enemy; while the monster in the middle is the boss.

After certain amount of enemies spawn in the scene, a boss enemy will be instantiated.

Fig. 13, Game Scene - Spookiland and enemies

Each enemy and boss will have their own animation for walking and dying. For example, the pumpkin man and skeleton will walking like a zombie, and fall down when dying, which matches the environment and their look.

iv. Weapon

We have imported and integrated different types of weapon into the game. Players can choose from Bullet, Dark Missile, Fire Burn, Fire Rock, Ice Wave, Knife, Lightning Fissure. Only 2-3 types are available in each stage to fit the theme.

Fig. 14 shows the effect of Lightning Fissure in Planet 404 on the left; and the effect of Fire Rock in Snowman Island on the right.

Fig. 14, Gameplay scene with different weapons

v. Presents

Presents will appear as a treasure chest randomly (as shown in Fig. 15), if player hit the chest, a random reward among 3 types of available rewards (Fig. 16) will be given.

Fig. 15, a treasure chest containing reward appear in the scene

Fig. 16, list of available rewards

The available rewards are Shield, Half Damage and Medicine (from left to right). For Shield, a semitransparent sphere will render, to surround and protect the player, no enemies can pass through it (as shown in Fig. 17). For Half Damage, the hit of enemies will be decreased by half. For Medicine, it refills the health point of player by 20.

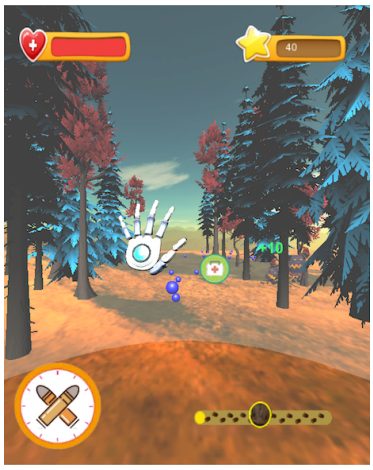


Fig. 17, Shield is triggered to protect player

vi. UI Elements

Different UI components are designed to show information to the player, such as HP Bar, Score, Weapon icon, boss progress bar and buttons (as shown in Fig. 18).

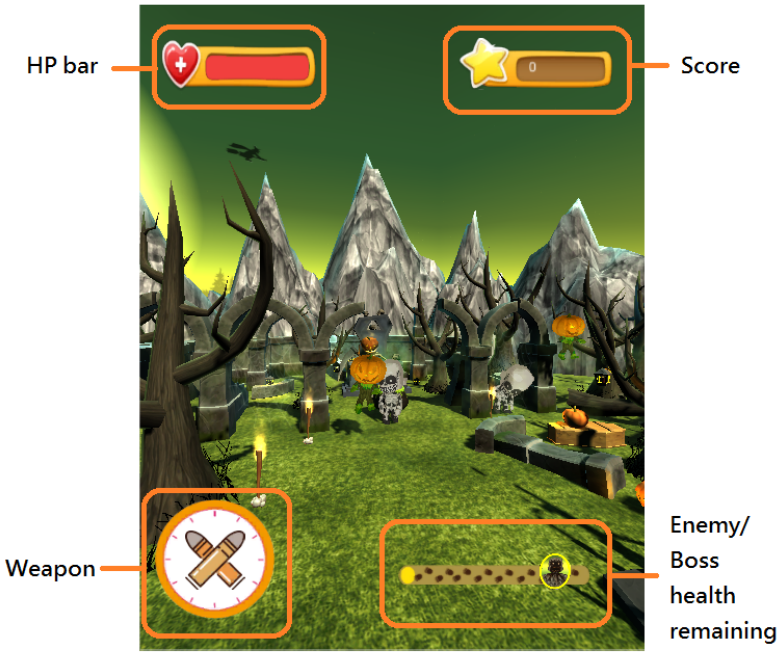


Fig. 18, Horror scene with UI elements indicated

At first, two sets of the same components were put into the left

and right side of the canvas manually to fit into the VR vision. And now, we changed to select “Screen based-camera” mode of canvas rendering mode, hence, the UI components can fit the VR camera in a more accurate way.

2.2.3 Hardware Design

Recall of Fig. 9 from Design section, apart from the Main Controller (the software part), the head-mounted display, Leap Motion and mobile phone are the core components to allow players to enjoy VR game.

To implement the hardware design, Google Cardboard, as shown in Fig. 19, was firstly chosen. Cardboard utilizes the screen of smartphone and split the image into two. It is accessible to general players since it is much cheaper than other VR gadgets such as Oculus Rift and HTC Vive. Everyone can have a taste of VR with his or her smartphone. However, after a few trials and User Acceptance Tests, Google Cardboard was found to be unsuitable because of the absence of headband to mount itself onto player’s head and insufficient space for players with glasses. Therefore, we picked another VR head-mount, VR Box, which meets our requirements with a comfortable design, as shown in Fig. 20.

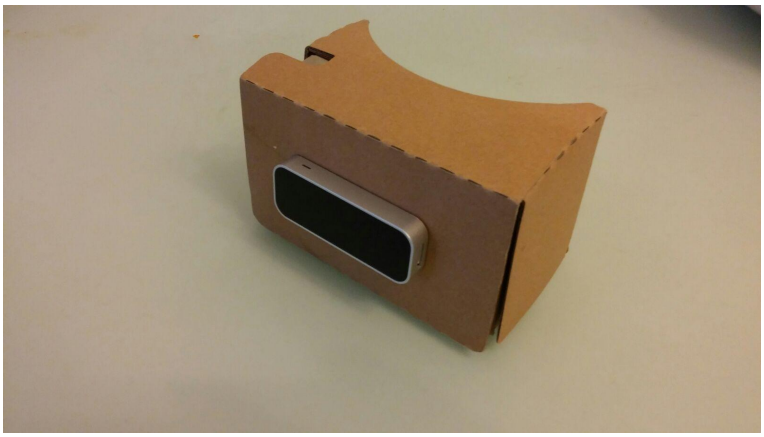


Fig. 19, Project’s VR camera setup with Google Cardboard

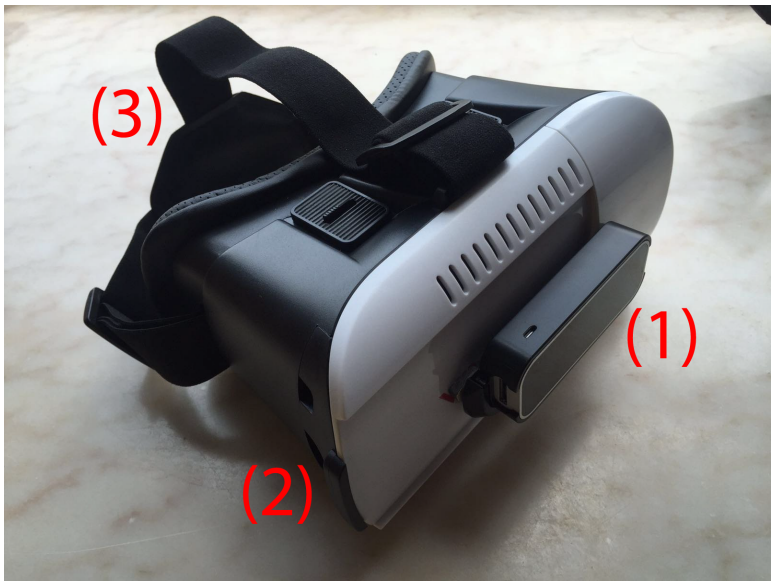


Fig. 20, Project's VR Camera setup with VR Box

Leap Motion is mounted in front of the VR Box at position (1). It ensures player's hands are accurately captured no matter where he or she is facing.

Mobile phone is clamped by a slider and embedded into the VR Box at position (2). Players are able to view the mobile phone screen through the lens of the VR Box.

The bandage at position (3) is the headband to mount itself onto player's head. It is one of the reasons why we prefer VR Box to Google Cardboard.

2.2.4 Software Architecture

i. Hand Store and Shooting Manager

Referencing to Fig. 10 from the Design section, Hand Store and Shooting Manager are responsible for receiving hands' data, recognizing hand gestures and managing shooting behaviors. Players are able to shoot bullets from his palm by opening his hand. The technical details are as follow (as shown in Fig. 21).

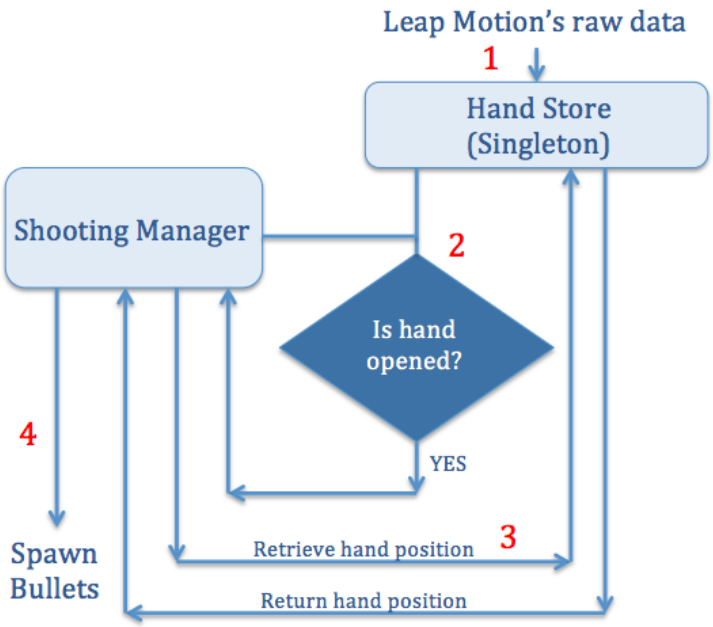


Fig. 21, Designed implementation of shooting bullets

1. Leap Motion's raw data input into HandStore, which is a singleton to abstract all hand-related information.
2. After the game starts, ShootingManager will query through HandStore's APIs to know whether the hands meet the shooting criteria, for example, hands opened (Fig. 22).
3. If the criteria is met, ShootingManager will ask HandStore for hand position, rotation and velocity information.
4. After retrieving hands information from the HandStore, ShootingManager will be responsible for spawning bullets on the screen.

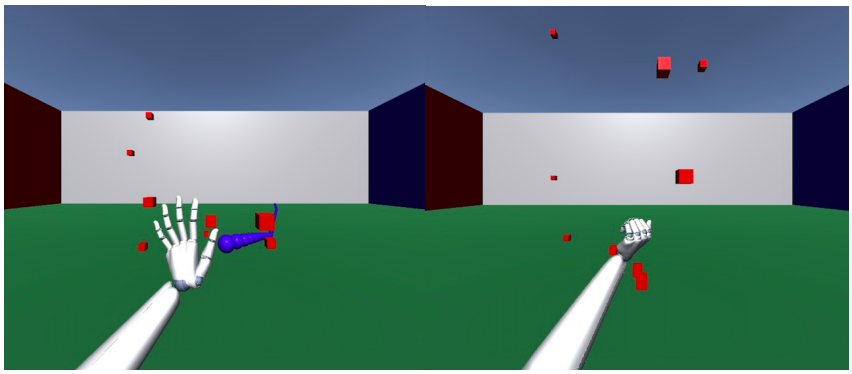


Fig. 22, Start shooting bullets when hand is opened (left), stop shooting and change bullet type when the play is shaking a closed-hand (right)

Leap Motion's raw data are provided by HandControllerCycler game object from Leap Motion SDK. HandController script from HandControllerCycler game object provides an API to access the Hand Models.

There are 4 main classes in Leap Motion's skeletal tracking model [19], they are "HandModel", "Hand", "Finger" and "Bone", each one is the subset of the previous one. Every class provides sufficient API to access the state, position, orientation, velocity, etc. For details of the APIs provided by Leap Motion SDK, developers can checkout the reference [20].

HandStore is designed to ensures data from Leap Motion SDK are abstracted from other classes in our system. Hands' data analysis algorithms are encapsulated in HandStore, only certain APIs can be called from outside, such as, IsHandOpened(). It gives ShootingManager a convenient way to know whether bullets should be spawned, but the details of how to detect gesture programmatically are abstracted from other classes.

ShootingManager is also able to change weapon when the player shakes a closed-hand (Fig. 22). The ShootingManager in each stage will define a list of weapons (Fig. 23) that the player can use. Referring to Fig. 23, weapons from left-to-right and top-to-bottom are Knife, Bullets, Fire Burn, Lightning Shot, Fire Rock, Ice Wave, Lightning and Dark Missile. When the gesture is detected, next weapon will be selected as the current weapon.

Fig. 23, List of available weapons

HandStore is able to recognize 2 types of hand gestures.

1. An opened-hand
2. A swiping closed-hand

To detect an opened-hand, HandStore will loop through the 5 finger models and query IsExtended() API to see if all the fingers are extended. Fig. 24 shows the code snippet.

```
// check if hand is open
public bool IsOpenHand(HandModel handModel){
    int extendCount = 0;
    Hand leapHand = handModel.GetLeapHand ();
    foreach (Finger finger in leapHand.Fingers) {
        if (finger.IsExtended) {
            extendCount++;
        }
    }
    return extendCount >= 5;
}
```

Fig. 24, code snippet for detecting if hand is opened

To detect a swiping closed-hand, HandStore will loop through the 5 finger models and query IsExtended() API to see if all the fingers are not extended. And followed by checking whether the velocity of the hand is greater than a threshold value. If yes, it is considered as a shaking closed-hand. Fig. 25 shows the code snippet.

```
// check if hand is closed
public bool IsCloseHand(HandModel handModel) {
    int extendCount = 0;
    Hand leapHand = handModel.GetLeapHand ();
    foreach (Finger finger in leapHand.Fingers) {
        if (finger.IsExtended) {
            extendCount++;
        }
    }
    return extendCount <= 0;
}

public float PalmVelocity (HandModel handmod) {
    Hand leapHand = handmod.GetLeapHand ();
    Vector handSpeedVector = leapHand.PalmVelocity;
    return handSpeedVector.Magnitude;
}
```

Fig. 25, code snippet for detecting if hand is closed and shaken

To implement change weapon feature, parameters such as "bullet prefab", "number of bullets" and "weapon icon" have to be changed. And it is logical that hand gesture should be detected by ShootingManager and change weapon parameters should be initialized from player rather than from weapon itself. Fig. 26 illustrates the logic flow.

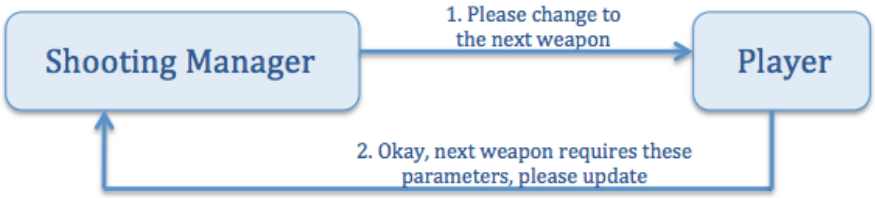


Fig. 26, logic flow of changing weapon

However, it is not scalable to store all the required parameters in Player class. To implement scalable software, strategy pattern is chosen. Each Player "has a" weapon behavior, every weapon inherits from abstract class IWeapon and implements a function called "SetShootingController". This allows a Player to pick a Weapon, and directly call "SetShootingController" function to receive all the parameters that are needed for ShootingManager. Fig. 27 illustrates the design pattern and Fig. 28 is code snippet of weapon behavior.

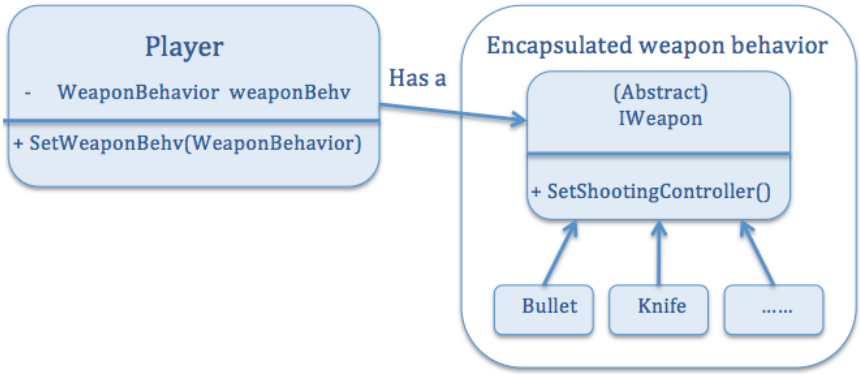


Fig. 27, strategic pattern of weapon implementation

```
public interface IWeapon {
    WeaponManager.WeaponType WeaponType ();
    void SetShootingController ();
}

public class BulletBehv : MonoBehaviour, IWeapon{
    public WeaponManager.WeaponType WeaponType () {
        return WeaponManager.WeaponType.Bullet;
    }

    public void SetShootingController (){
        ShootingController shootControler=  ObjectStore.FindShootingManager ().GetComponent<ShootingController>();
        shootControler.bulletPrefab = (Object)Resources.Load("Bullet") as GameObject;
        shootControler.numOfBulletPerSecond = 10;
        shootControler.UpdateWeaponIcon (WeaponManager.weaponIconBasePath + "weapon_bullet");
    }
}
```

Fig. 28, code snippet of weapon behavior

ii. Enemy Manager

EnemyManager is responsible for spawning enemies and managing spawning parameters, such as monster types, spawning speed, and maximum enemy number.

Number of enemies is monitored by EnemyManager. It spawns new enemies when the enemy count has not reach the upper limit. As illustrated in Fig. 29, in each updating cycle, EnemyManager will check whether the count reaches upper limit. If it does not reach, one enemy will be spawned in a random position around the player. Once the count reaches upper limit, EnemyManager will spawn a boss. Killing the boss marks the end of the stage.

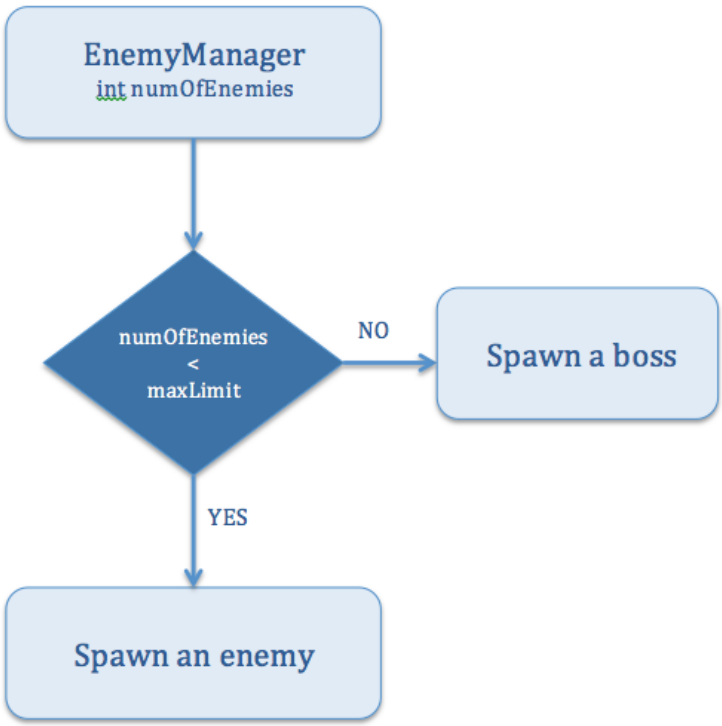


Fig. 29, Designed implementation of spawning enemies

Also, enemies are stage-specific. EnemyManager keeps a unique list of enemies in each scene. For example, Planet 404 is having alien as enemy and Snowman Island is having snowman as enemy, as shown in Fig. 30.

Fig. 30, snowman & Santa enemy in Snowman Island

In each updating cycle, if an enemy needs to be spawned, EnemyManager will randomly select an enemy from the list of enemy options and spawn it.

Enemies are divided into many classes, every enemy has its own class and with the hierarchy shown in Fig. 31.

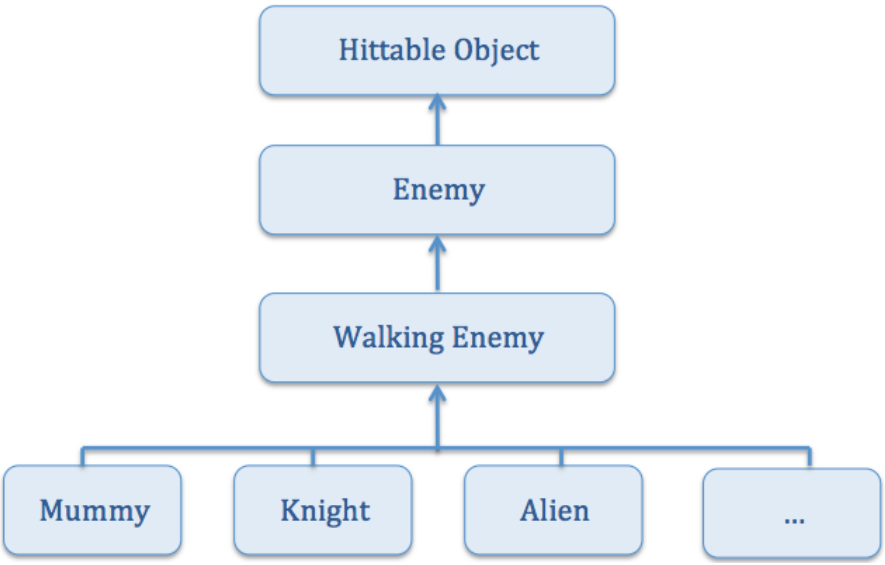


Fig. 31, software architecture of enemy classes

Hittable Object class controls game object's health and state of living. This class is designed for all game objects with life to

inherit from.

Enemy class manages attributes of an enemy. These include moving speed, moving direction, boundary region, hit sound, etc. This is the most important class that defines how enemy response to bullets, move and attack player.

Walking Enemy class handles paddling degree when the enemy is walking from spawning position to destination.

Enemy-specified classes are mainly designed for handling animations. As different enemy has different animation, these classes are responsible for initializing the animations and some specific transform actions.

As discussed in Model Design (Section 2.2.2), when spawning an enemy, there is certain probability that a present will also be spawned. By shooting the present, special effects will be redeemed. Fig. 32 shows the hierarchy of Present class.

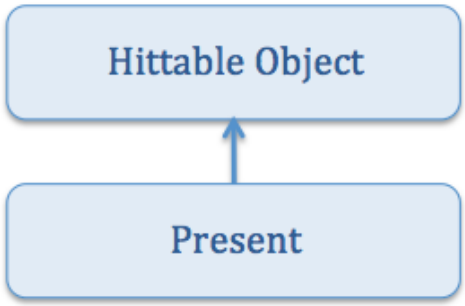


Fig. 32, software architecture of enemy classes

Present is also inherited from Hittable Object, it is designed to be a hittable object with 1 HP, which means it will be destroyed whenever a bullet hits. Strategy pattern is also selected to implement the effects of present. During present initialization, a present behavior is randomly chosen and set, i.e. each present "has a" present behavior. Every present behavior inherits from abstract class IPresent and implements a function called "RedeemEffect". This allows a Present to randomly pick a present behavior during initialization and call "RedeemEffect" function when it is being hit. Fig. 33 illustrates the design pattern and Fig. 34 is code snippet of weapon behavior.

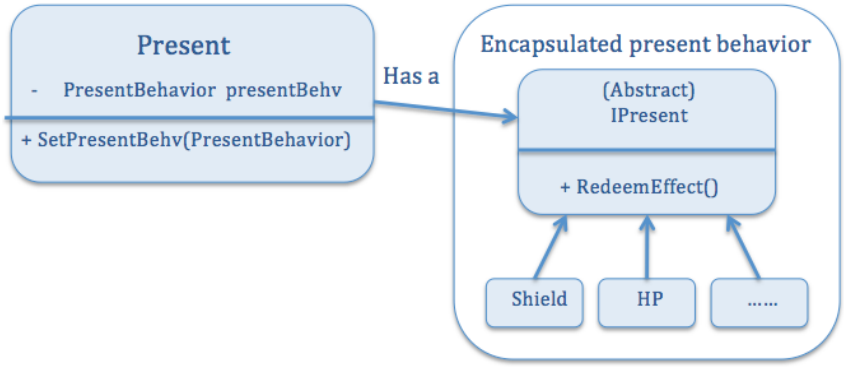


Fig. 33, strategic pattern of present implementation

```
public interface IPresent {
    PresentManager.PresentType PresentType ();
    string GetPresentIcon ();
    void RedeemEffect (Player player);
}

public class ShieldBehv : IPresent {
    public PresentManager.PresentType PresentType () {
        return PresentManager.PresentType.Shield;
    }

    public string GetPresentIcon () {
        return PresentManager.presentsIconBasePath + "shield";
    }

    public void RedeemEffect (Player player) {
        Debug.Log ("Redeem effect: Shield");
        SpecialEffect.AddShield (player);
    }
}
```

Fig. 34, code snippet of present behavior

iii. Network Manager

In Network Manager, we have customized network configuration for PUN server connection and abstraction of PUN functions for our project use. Main controller can set up connection to PUN server by calling utilities in Network Manager. Game rooms are set with a suitable synchronization rate and have limit on number of existing player.

Apart from back-end connection setup, we also implemented the in-game synchronization of information and game objects. By attaching PhotonView component to gameobject, transform data such as position and rotation of the object can be synchronized through PUN. However, synchronizing every game objects in scene will put a great burden on the game's

traffic handling. Large traffic data will be generated as the transform data of objects are updated every frame. It could cause low frame rate per second (FPS) and high latency. Therefore, the game has to strike a balance between fully synchronization and performance.

In order to cater the bandwidth problem, a hybrid approach between PhotonView and remote procedure call (RPC) is used. RPC are function calls that initiate from one client and broadcast to other clients through server. In our game, player and enemy objects are sync with PhotonView as the number of these objects are bound and highly subjected to user's control, which requires precise synchronization. On the other hand, bullets spawning are controlled by RPC. When Shooting Manager try to shoot a bullet, information including spawn location and bullet velocity is passed to a RPC function. The function is then triggered in all clients and spawns a "local" bullet, without PhotonView. As all parameters of the bullet are the same when spawn, identical movement is expected under the same physics engine and environment. A large amount of Internet traffic is saved as there could be up to a hundred bullets co-existing in one instance if all players are shooting at the same time.

We also implemented single player mode by utilizing PUN offline mode setting. Offline mode enable access to certain PUN functions and RPC calls without actually connecting to the game server. It allows us to reuse the network-aware code work perfectly under single player setting.

iv. Scene Manager

Scene Manager is a wrapper of Unity default SceneManager. It maps all the scene indices and only performs one job - change scene. Whenever need to change to another scene, a scene index must be passed into Scene Manager, and it will perform validation and customization. For example, if the scene index passed in is ScoreScene, Scene Manager will retain the decoration of the current scene. Fig. 35 shows the code snippet.

```
public const int MENU_SCENE = 0;
public const int BRIGHT_SCENE = 1;
public const int TUTORIAL_SCENE = 2;
public const int SCORE_SCENE = 3;
public const int MAP_SCENE = 4;
public const int SPACE_SCENE = 5;
public const int CHRISTMAS_SCENE = 6;
public const int HORROR_SCENE = 7;

public static void ChangeToScene (int sceneIndex) {
    if (sceneIndex == SCORE_SCENE) {
        RetainDecoration ();
    }
    SceneManager.LoadScene (sceneIndex);
}
```

Fig. 35, code snippet of Scene Manager

v. Scoreboard and Statistic Store

Scoreboard and statistic store are utilities that track and display data for user. Although both scoreboard and statistic store are handling statistical data, the architecture are different.

Scoreboard works with PUN PhotonPlayer library, which help storing players' name and score. Unlike other statistical data, the score display on screen has to be constantly updated. However, updating by a fixed frequency is not ideal. Short frequency would cause many unnecessary update while long frequency may cause a delay. Therefore, scoreboard is implemented with observer pattern. Score display in game scene (top-right corner) is registered in scoreboard as an observer. When score change, scoreboard will broadcast an score update to all observers. On-demand update helps to reduce burden especially during play time.

On the other hand, statistic store is implemented with singleton to make sure only one instance of statistic store exist throughout the game. It keeps track of game start/end time, player max/end health, boss spawn/die timestamp, and enemy killed. These statistics are analyzed into play time, boss kill time, health loss, etc., and displayed to the player at the end of each game, i.e. in score scene.

2.3 Testing

There are differences between game testing and common software testing. Game testing is not only having test cases to ensure the logic work, but needs more tests on unexpected behavior from user interaction, graphical burden, and network.

The development is divided into different modules. To ensure each part is reliable, modules have been tested in a separate project to make sure it is working and the integration effort is minimized.

2.3.1 Shooting

i. Ability

Testing for the gesture detection of Leap Motion is required. Leap Motion performance depends on the player's hand gesture and the placement of itself. Therefore, we need to check for a set of gestures that we are using in the project and evaluate the detection performance. Filtering out gesture with bad performance gesture can avoid player's frustration during the gameplay. It can be tested by observing the hand rendering under different gestures.

After thorough testing, we came up with two hand gestures that are mentioned above. An opened-hand is the most easiest gesture that can be recognized using Leap Motion. Also, it is intuitive to player that it is a shooting hand gesture. On the other hand, in order to prevent accidentally changing weapon during shooting, a closed-hand, a totally different hand gesture, is chosen to be a "change weapon" gesture. At the beginning, the gesture is too easy to be triggered, therefore, we changed our mind and required a shaking closed-hand to trigger "change weapon" action.

ii. Bullet Spawning

Bullets can be easily rendered by adding "ShootingManager" to the scene. "ShootingManager" takes care of which bullet type to be spawned, when the bullets should be spawned and in which direction bullets should go. This feature has also been tested to ensure shooting logic is scalable and will not cause huge integration effort whenever a new bullet type is introduced.

2.3.2 Enemy

Enemies can be easily spawned by adding an empty object prefab called “EnemyManager” into the scene. By inputting required parameters, such as spawning area, enemy prefabs, maximum number of enemies and spawning interval, enemies will be spawned automatically. The feature has been tested carefully to ensure whenever a new scene or a new monster is drawn, enemy spawning logic will not cause huge integration effort.

2.3.3 Networking

The project aims at creating a multiplayer game. As PhotonView only synchronize models transform data, other parameter such as health, score need to be sync correspondingly. Testing is required to ensure non-transformed data are correctly sync in different state of game. In this part of testing, we wrote scripts using OnPhotonSerializeView function provided by Photon Unity Networking package to synchronize blood level, bullet type information in a 2 players environment.

2.3.4 Scene rendering

Scene rendering can cause great burden to the GPU and hence affecting the performance of the game. To be aware of the limit of GPU rendering, scenes with different number of objects, texture quality and level of details have been tested to give developers a brief understanding of how much details can be added.

In our early development, Fig. 36 forest scene is used as our game default background.

Fig. 36, Early forest scene

The details of tree and leaf models are incredible and the frames per second (FPS) performance is smooth when played in Unity editor. However, after we build and export the project as a standalone program, the performance drop significantly to single-digit FPS. It is caused by implicit graphic downgrade in editor, which is used to reduce graphical workload during development. Therefore, when the project is built, the graphic quality reset to the original level and result in low fps. To alleviate the bad performance, we designed several forest scenes with different levels of detail (as shown in Appendix A) and finally chose one with other art style that has fewer burdens to the GPU while the appearance is still attractive.

2.3.5 Leap Motion detection Range

Leap Motion used infrared radiation (IR) to detect hand at the front. Because of this nature, Leap Motion can only detect precisely within a certain area. If player' hand is too close, the detector is mostly covered and cannot recognize hand's shape. If it is too far, the IR is too weak for clean recognition. It is very important to have clear idea of detection range of Leap Motion in order to produce a positive game experience. To test that, we mount the Leap Motion to the VR head-mount and launch a Unity project with Leap Motion enabled. Then we check for the range of hand model rendering with X, Y, Z axis.

After testing, we discovered that Leap Motion has options to increase the detection range up to 25 cm and enable "Low Resource" mode to reduce performance and bandwidth of the Leap Motion Controller to improve reliability on slower computers.

2.4 Evaluation

Apart from the evaluation by project team, User Acceptance Test (UAT) has been conducted in order to collect players' feedback and make enhancements to our game. 5 players have been invited to the UAT, and we have observed their performance and asked them about their opinions and feedbacks. The tests mainly focus on the following aspects:

2.4.1 Precision

Precision is a very crucial factor that affects the gameplay experience of a FPS game. If players cannot control or perform attack precisely, they may feel frustrated during the game. Therefore, players have been asked questions such as if they can shoot the target accurately and whether the hand gesture can be detected correctly.

As we observed, most players performed the shooting action precisely. Although 1-2 of them were unable to shoot, as they place their hand too close to the leap motion, after some trials, they were able to play it and shoot precisely. It is a reasonable result, and we have added some more instruction in the tutorial scene to ensure players can play it successfully.

2.4.2 Smoothness

A lot of 3D models rendering and bullet and enemy spawning are included in this project. Together with network element and object synchronization, the game could create a huge burden to the computer and affect game's smoothness. Information such as whether players can shoot fluently or players experience any lags is valuable for the refinement of our project.

There are no serious lags during the game play overall. A problem of the hand from Leap Motion was stucked in the scene occasionally during UAT and the testing from our team. We spent some time and eventually solved the problem. After solving this problem, the smoothness of the game is good.

2.4.3 Intuitiveness & Ease of use

If a game is able to played by a person smoothly before he or she has gone through the tutorials, it implies that the game flow and user interface are well-designed such that users can play it with their intuition. In this evaluation, players are asked to play it once with the entire tutorial skipped, and then give their reflection on whether the game is easy to control.

When players were asked to play directly without any tutorial and instruction, the main feature of the game is still intuitive, which is to open hand and shoot the enemies. However, other features like changing weapons are not intuitive that most of them just changed it accidentally or did not notice it. After tutorial scene, most players played and handled all the features, and they agreed that it was easy to use. The result showed that the game is easy to control, and tutorial improves players' experience a lot.

Evaluations were performed throughout the development process, which is shown in Section 3.2 GANTT Chart, improvements on the project were made based on the users' feedback.

3. DISCUSSION

The objectives of our project is to achieve a more realistic VR gaming environment by utilizing Leap Motion as control input. Meanwhile, we would also add multi-player element into our game as many of the current VR are focusing on single player. Lastly, we would like to accomplish our target with a more accessible price to let more people to experience VR gaming. Based on these objectives, we think we have successfully achieve our targets.

At the beginning of the project, none of us have VR gaming experience as the VR-enable setup and devices are quite expensive. We explored different way to build our game, such as whether we should develop it as a mobile game or PC game, or should we use USB or wifi for streaming. At last, we settled with developing a PC game and streaming the screen from PC to Android device through the Trinus approach.

For game themes, we chose a rather simple style instead of a realistic one. Realistic style often associated with small details and high quality texture. Using a cartoonish theme reduce the burden on graphic generation and streaming, and save us a lot of time and effort on model and UI design. And we have tried using realistic style and it was lagging.

Initially, we planned to have other enemy types, for example flying enemy and enemy that can shoot bullet. Due to time limitation, we only implement one major enemy type, which is walking enemy. However, in the architecture design (Fig. 31) of enemy classes, we prepared a good design to scale up and implement in the future.

Throughout the whole project, we have conducted multiple UAT regarding ease of control and game flow, which lead to several extra features.

- **Tutorial scene**
  - Due to Leap Motion limitation, hand may not be detected when it is too close to the device. We noticed that some users fail to play game because of that. Therefore we add a tutorial session before the play time start. It teaches users to play by step-by-step guidance with instructions. Users can only proceed when they fulfill the criteria given by the system

- **Boss fight**
    - Part of the users mentioned although the higher level could be challenging, it is sometime repetitive and they never know when would it ends. We therefore added the progress bar showing how many enemies remaining, and a boss fight afterward. It gives a clearer indication of progress and a remarkable end.

During our development, we put extra attention on designing the architecture of components and the interaction between them. We implemented hierarchy of object classes for different objects in game scenes. New types of enemy or environmental objects can be added easily by inheriting corresponding class. We also apply appropriate design patterns for different modules, for example, strategy pattern for weapons and presents. These measures are taken to ensure our game has a high scalability.

4. CONCLUSION

In this project, we completed a VR first-person-shooting game controlled by Leap Motion. Users are able to enjoy the game without using keyboard and mouse but direct hand gestures. Moreover, players are able to play and have fun with their friends in the same instance.

Although the game functions perfectly at the moment, there are room for improvements to make the game more interesting.

1. Enemy types

   Currently there is only one enemy type, the walking enemy. This enemy would only walk towards the player and do melee attack. More enemy types, such as flying enemy and shooting enemy can make the game more challenging and less repetitive.

2. Hand gestures

   We have one gesture for shooting and one for change weapon. More gesture detections can be added. At the initial stage, we planned to have other gesture for functionality like one-hand shield. However, as we explore Leap's API, some gestures are hard to detect due to noisy data. If we have more time, we can explore more gestures with high accuracy.

3. Multiplayer interaction

   Our game allows multiple players play in the same instance but with limited interaction. New settings which encourages users to cooperate with each other, or a Player-VS-Player (PvP) game mode can be added to the game.

Our team has learnt a lot of things through the final year project, from technical to interpersonal skills.

1. Project management and development

   We used various development tools in this project; for example, we used Trello for issue management and Github for source code management. We also had a better understanding to project development lifecycle, how to design a feature and divide a feature into small parts that are manageable.

2. Coding skills

   Multiple design patterns are implemented in the project, such as singleton classes, observer pattern, strategy pattern, etc. This shows that we have applied some best practices we learned to our project.

3. Communication skills

   During the development process, we came across reviews and tests that involve people who do not have technical background. It is crucial that we are able to explain complicated technical concepts to these people. In this project, we have practice a lot on communication skills.

This is an unprecedented opportunity for us to dedicate several months on one project, from researching topic to finishing the project. Game development involves various skillsets and requires a lot of work. We have gained solid experience on splitting workload, synchronizing and merging each part. These experience and skills are very crucial and useful in our future career.

We have already made our repository on Github public to everyone. We hope it would help our fellow students to learn more about game development, VR, and Leap Motion, or even consider putting these elements into their future projects.

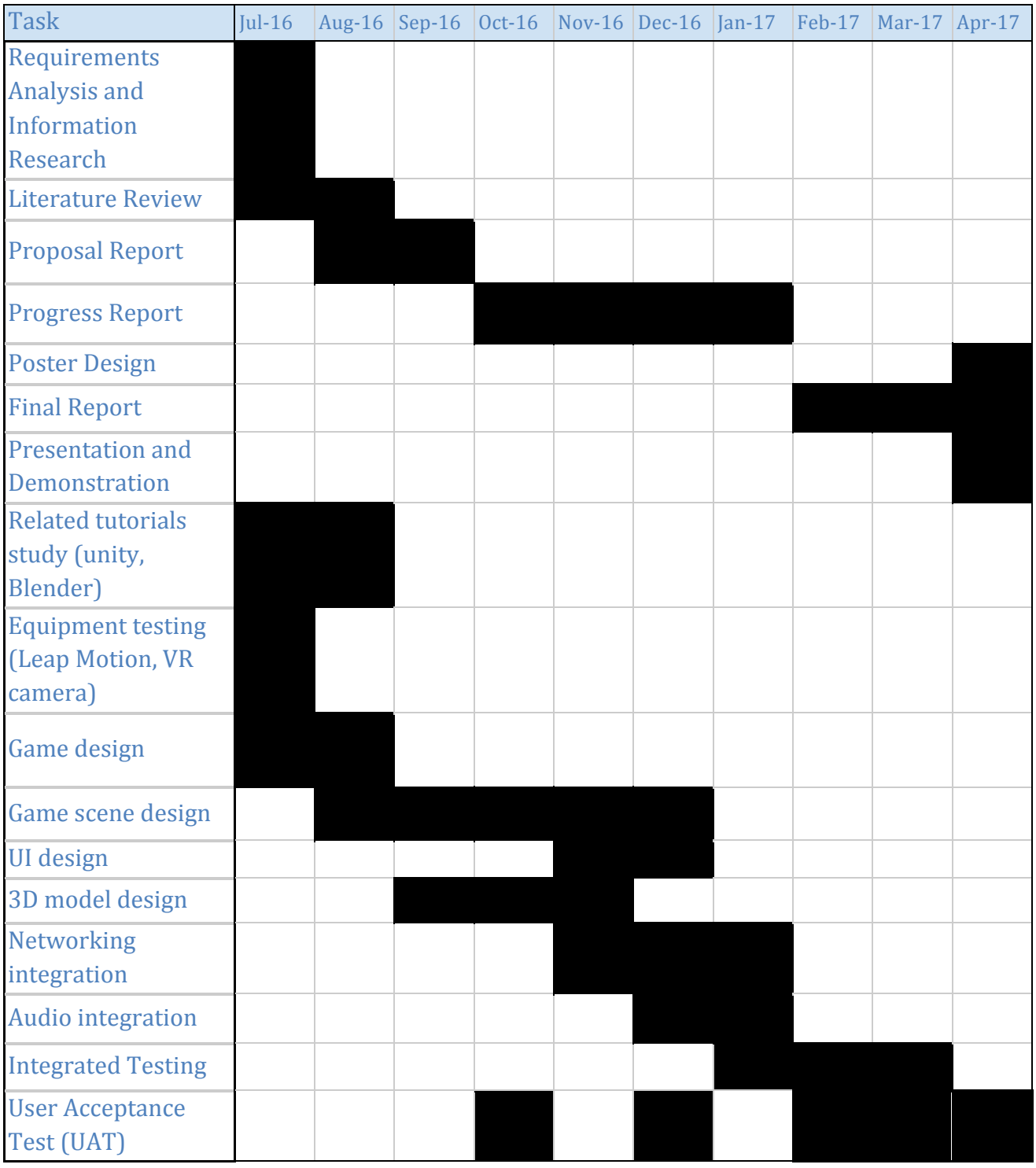
5. PROJECT PLANNING

5.1 Division of Work

A = Assistant
L = Leader
x = Work together

Task	Alice Sin	Johnny Wong	Jeffrey Lee
FYP project related			
Requirements Analysis and Information Research	x	x	x
Literature Review	x	x	x
Proposal Report	x	x	x
Progress Report	x	x	x
Poster Design	x	x	x
Final Report	x	x	x
Presentation and Demonstration	x	x	x
Game structure related			
Related tutorials study (Unity, Blender)	x	x	x
Equipment testing (Leap Motion, VR camera)	A	A	L
Networking integration	A	L	A
Audio integration	L	A	A
Repository management	A	A	L
Integrated Testing			
Server test	L	A	A
Network latency test	L	A	A
Motion recognition test	A	L	A
User interface test	A	A	L
Game logic and flow test	A	L	A
Game design related			
Game scene design	A	L	A
UI design	A	A	L
3D model design	L	A	A
Game design	x	x	x

5.2 GANTT Chart



6. HARDWARE AND SOFTWARE REQUIREMENTS

6.1 Hardware Requirements

PC with Operating System: Microsoft Windows	Purpose: Hosting
Leap Motion	Price: approx. HK$ 620.40 (US$ 79.99) [18] Purpose: Hand motion detection
VR Box	Price: approximate HK$ 90 (US$ 11.60) Purpose: VR head-mount for game experience
Android phone	Requirements: Android version 4.0 or above Purpose: Gaming device, use with Google Cardboard as VR head-mount

6.2 Software Requirements

Unity	Requirements: Version 5.3.4f1 or above Purpose: Game development
Trinus VR	Price: HK$ 64.49 [21] Purpose: Streaming screen from PC to phone and detect head movement
Adobe Photoshop, Illustrator	Price: HK$ 148/month per application [22] Purpose: Graphic design for game and user interface
Git	Purpose: Version control
Blender	Purpose: 3D Game object design

7. REFERENCES

[1] Robert Workman, “A Brief History of Video Game Virtual Reality, and Why This Time Will be Different”, *Prima Games*, April 2014; https://www.primagames.com/games/killzone-shadow-fall/feature/brief-history-video-game-virtual-reality-and-why-time-will-be-different

[2] Nick Pino, “GDC 2016: Everything we saw from Sony, Microsoft, Oculus and HTC”, *techradar*, 19th March 2016; http://www.techradar.com/news/gaming/gdc-2016-what-we-want-to-see-from-sony-microsoft-oculus-and-amd-1315725

[3] Exit Games, “Photon Unity Networking Free on Unity”, *Unity3D Asset Store*, 23rd Aug 2016; https://www.assetstore.unity3d.com/en/#!/content/1786

[4] Superdata, “Virtual Reality Industry Report 2016”, *Superdata*; https://www.superdataresearch.com/market-data/virtual-reality-industry-report/

[5] Knero, “Virtual Reality Games by Genre – the Q3 KZero VR Software Radar”, *Kzero Worldwide*, 2014; http://www.kzero.co.uk/blog/virtual-reality-games-by-genre-the-q3-kzero-vr-software-radar/

[6] Barnabás Takács, Lajos Simon, “A Clinical Virtual Reality Rehabilitation System for Phobia Treatment”, *IEEE Xplore*, 2007; http://ieeexplore.ieee.org.lib.ezproxy.ust.hk/stamp/stamp.jsp?tp=&arnumber=4272069

[7] Sim Lee Sen, Yong Bang Xiang, Eileen Su Lee Ming, Yeong Che Fai, Khor Kang Xiang, Qamer Iqbal Khan, “Enhancing effectiveness of Virtual Reality Rehabilitation System: Durian Runtuh”, *IEEE Xplore*, 2015; http://ieeexplore.ieee.org.lib.ezproxy.ust.hk/stamp/stamp.jsp?tp=&arnumber=7244607

[8] Statista, “Mobile game in Hong Kong - User by age group”, *Statista - Digital Market Outlook,* 2015; https://www.statista.com/outlook/211/118/mobile-games/hong-kong#market-ageGroup

[9] fast Company, Download page of “Lost in the Kismet”, *Google Play*; https://play.google.com/store/apps/details?id=com.hihill.link

[10] Com2Geek, Download page of “VR Bike”, *Google Play*; https://play.google.com/store/apps/details?id=com.poly.VRBike

[11] Bob Vlemmix, “Manus VR - Welcome to Pillow’s Willow”, *Manus VR Blog*, 5th February 2016; https://manus-vr.com/05022016article

[12] Kickstarter, “A Brief History of Oculus, from Day Zero to Day One”, *Medium*, 6th Jan 2016; https://medium.com/kickstarter/a-brief-history-of-oculus-from-day-zero-to-day-one-8878aae002f8#.sw2n74tre

[13] Lily Prasuethsut, “HTC Vive: Everything you need to know about the Steam VR headset”, *Wareable*, 23rd February 2016; http://www.wareable.com/vr/htc-vive-vr-headset-release-date-price-specs-7929

[14] Official website of Leap Motion; https://www.leapmotion.com/

[15] Alex Colgan, “Android SDK Hardware Requirements”, *Leap Motion Community*, 4th April, 2015; https://community.leapmotion.com/t/android-sdk-hardware-requirements/2845

[16] Official website of Trinus VR; http://trinusvr.com/

[17] Alex Colgan, “Leap Motion Inc. World’s First VR Dance Compeition”, *Leap Motion Community*, 18th August, 2016; http://blog.leapmotion.com/worlds-first-vr-dance-competition/

[18] “Leap Motion Controller”, *Official website of Leap Motion*; http://store-us.leapmotion.com/products/leap-motion-controller

[19] “Introducing the Leap Motion Skeletal Tracking Model”, *Official website of Leap Motion*; https://developer.leapmotion.com/documentation/javascript/devguide/Intro_Skeleton_API.html

[20] “Leap Motion SDK API Reference”, *Official website of Leap Motion*; https://developer.leapmotion.com/documentation/javascript/api/Leap_Classes.html

[21] Odd Sheep, Download page of “Trinus VR”, *Google Play*; https://play.google.com/store/apps/details?id=com.loxai.trinus.full

[22] “Creative Cloud Price and Membership”, *Adobe Creative Cloud*; https://creative.adobe.com/zh-tw/plans?single_app=illustrator&promoid=YB1TGRTR&mv=other&store_code=hk

8. REFERENCES OF RESOURCES USED

Asset:

[1] Skybox for Menu - Cope! Free Skybox Pack, 70:30,
https://www.assetstore.unity3d.com/en/#!/content/22252

[2] Decoration for Menu and Present for all stages - Medieval Chest Pack, rereke,
https://www.assetstore.unity3d.com/en/#!/content/61923

[3] Weapon for Lost Forest - Crazt Weapon, Crazy Cube,
https://www.assetstore.unity3d.com/en/#!/content/15090

[4] Decoration for Lost Forest - Stylized Nature Pack, Mikael Gustafsson,
https://www.assetstore.unity3d.com/en/#!/content/37457

[5] Enemy in Lost Forest - Chibi Mummy, Richchet,
https://www.assetstore.unity3d.com/en/#!/content/60462

[6] Enemy in Lost Forest - Toon Death Knight Pack, Mesh Tint,
https://www.assetstore.unity3d.com/en/#!/content/48188

[7] Skybox for Planet 404 - Star Nest Skybox, Ninjapretzel,
https://www.assetstore.unity3d.com/en/#!/content/63726

[8] Decoration of Planet 404 - Vast Outer Space, Prodigious Creations,
https://www.assetstore.unity3d.com/en/#!/content/38913

Music:

[1] Background Music of Menu - Cute, Bensound.com
http://www.bensound.com/royalty-free-music/track/cute

[2] Background Music of Tutorial - The Lounge, Bensound.com
http://www.bensound.com/the-lounge

[3] Background Music of Snowman Island - Christmas Theme Music, dl-sound,
https://www.dl-sounds.com/royalty-free/category/holiday-season/

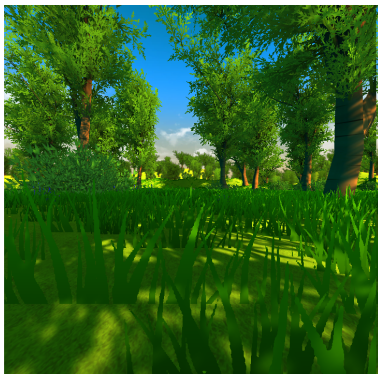
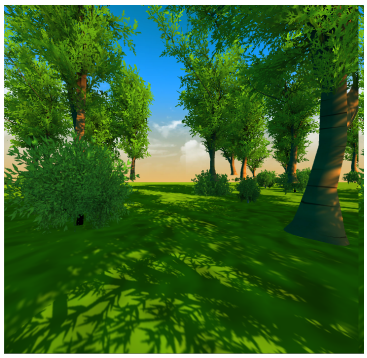
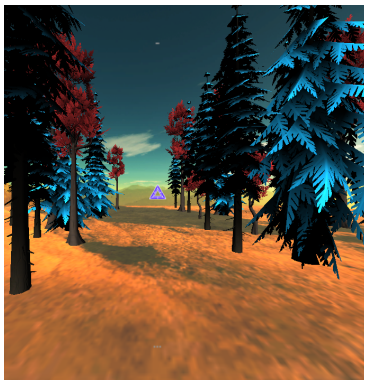
[4] Background Music of Planet 404 - Librarians in Space, Looperman.com,
https://www.looperman.com/loops/detail/102550/librarians-in-space-by-lankframpard-free-78bpm-hip-hop-synth-loop

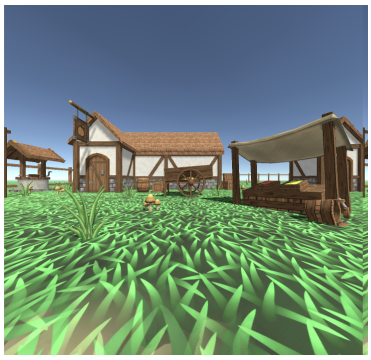
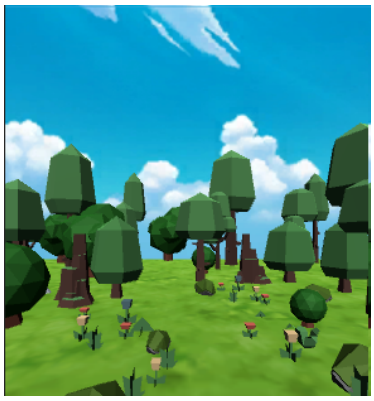
[5] Background Music of Spookiland - Children of Shadows, Playonloop.com,
https://www.playonloop.com/2016-music-loops/children-of-shadows/

Design Elements:
Most of the design are based on design from Freepik (http://www.freepik.com)

APPENDIX

APPENDIX A: RESULT OF COMPLEXITY TESTING

Sample Scene	Description	Level of Complexity	Result
	Very realistic, with detailed trees and grass and effect	Very High	Experience lag when start shooting
	Remove background terrain, small grass and wind effect	High	Smooth, but experience lag after export and under network condition
	Still realistic, but with less complex tree	High	Smooth even in network condition, with shooting and enemy animation

	Only complex for buildings	Medium (vertices: 600-3700)	Smooth even in network condition, with shooting and enemy animation
	Only low poly prefabs are used and no special effect on wind/ lighting	Low	Smooth in enemy spawning, shooting and networking

APPENDIX B: MEETING MINUTES

Minutes of the 1st FYP meeting

Date: 12th July, 2016
Time: 2:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Introduction to the FYP
    a. Desmond gave sets of 2-page briefing notes to students. Notes contain links of FYP and project research related information
    b. Desmond introduced FYP schedule, grading system and past FYP proposals
        i. FYP Grading system mainly divided into two parts, advisor (~60%) and reader (~40%)
        ii. Proposal report is the next item to be submitted online. Its deadline is on 22nd September, 2016
        iii. Introduced GANTT chart
    c. Desmond introduced HKUST CSE CT website for reference

2. Items discussed
    a. Google VR SDK
        i. Lee Chun On tested Google VR SDK on unity, but not demonstrated.
    b. Leap Motion
        i. Lee Chun On tested Leap Motion unity SDK on MacBook Air. Discovered that MacBook Air should use LeapMotionAssets version 2.3.1 at https://github.com/leapmotion/LeapMotionCoreAssets/tree/v2.3.1
        ii. Discovered behavior when using Leap Motion
            1. Two fingers stick together may be recognised as one finger
            2. Entering with the back of hand may be recognised as palm
            3. Detection range of Leap Motion cannot be too short (~5cm) or too long (~30cm)
            4. Items emitting infrared radiation may interfere Leap Motion's detection

5. Need to check if Leap Motion needs to be calibrated

c. Blender
   i. Sin Wing Lam tested Blender and drawn simple cubes to unity
   ii. 3D objects are advised to render in cubes, triangles and meshes that with fewer vertices
   iii. Maya can be used as it is well known in industries and it also has student friendly version
d. Game rendering methodology
   i. Unity generates iOS project which renders VR camera interface but cannot receive Leap Motion's information to render hands
   ii. Connect to computer and mirroring computer screen to device may be able to receive Leap Motion's information but cannot capture head tilting gesture
   iii. Decided to connect computer to screen directly and use mirroring technology to render game scene

3. Action Items
   a. Overview and objectives of project proposal
   b. Division of work and GANTT chart
   c. Try connect computer to screen directly and use mirroring technology to render simple game scene
   d. Check if Leap Motion needs to be calibrated

4. Meeting adjournment and next meeting
   a. The meeting was ended at 3:30 p.m.
   b. The next meeting will be on 12 July 2016 at 11:00 am in Room 3521

Minutes of the 2nd FYP meeting

Date: 26th July, 2016
Time: 11:00 a.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Sin Wing Lam

1. Methodology testing result

a. Our testing results of current streaming technology:
    i. Kinoni
        1. pros:
            a. Mapping any thing displayed in the screen of windows
        2. cons:
            a. Need to setup server
            b. Only for windows, not Mac
            c. Network lantency caused by Wifi connection
            d. Low resolution
            e. Head motion not detected
    ii. Trinus
        1. pros
            a. It supports USB tethering which has lower network latency
            b. Head motion detections are also available in Trinus environment
        2. cons
            a. Trinus application only support Android mobile devices
            b. Trinus SDK in Mac has problem
b. Decided to use Trinus for mapping game scene to mobile device and detecting head motions

2. Review on project proposal
    a. Overview & Objective
        i. Objective should address the following:
            1. Why the project worth doing?
            2. What are the pros and cons of existing methods?
            3. How we will make it better?
    b. Literature Survey
        i. Desmond suggested to have the pros and cons analysis of each tool that we selected in our project included (including Unity)
        ii. The pros and cons can be presented in table format
    c. Division of Work
        i. Having the documentation part marked as “Work Together” is ok
        ii. For some parts like Testing, Desmond encouraged us to have more details part (e.g. network, whole system)
        iii. Detail part can be divided into another separated table
        iv. Make sure division of work is even
    d. GNATT Chart
        i. Acceptable

3. Discussion on User Acceptance Test(UAT)
    a. Not compulsory for FYP but is a common practice in the industry
    b. Good to put it in our schedule for identifying bugs and gathering feedbacks
    c. Should be carried out throughout the project

4. Action Items
    a. Template of the entire project proposal
    b. Required hardwares and softwares
    c. Elaboration on Overview, Objective and Literature Survey
    d. Test the ways of combining all components (i.e. Leap Motion, VR camera, Phone & PC)
    e. Check if video demo compulsory for FYP

5. Meeting adjournment and next meeting
    a. The meeting was ended at 12:30 p.m.
    b. The next meeting will be on 3 August 2016 at 15:00 am in Room 3521

Minutes of the 3rd FYP meeting

Date: 3rd August, 2016
Time: 3:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Proposal draft of Introduction
    a. Document is fine
    b. Length is suitable and does not have strict constraint on length

2. Demonstration on leap motion, google cardboard and trinus VR SDK
    a. Wong Chun Yin demonstrated placing a phone that mirroring desktop game in google cardboard, with leap motion attached in front of google cardboard
    b. Handmade ribbon is needed to hold the component in front of eyes

c. Due to heavy loading, user's head may feel heavy and sometimes google cardboard pivot on user's nose causing pain
d. Dr. Desmond Tsoi suggested will attaching leap motion at the bottom of google cardboard feels better
e. Students suggest to replace google cardboard with some head-set that serve the same purpose, but more comfortable

3. Discussion on networking issue
   a. Dr. Desmond Tsoi suggested allowing distant player to join game is better, but local LAN is fine as well
   b. Unity support using NetworkManager class, but photon server is also a popular solution for networking in unity
   c. Network feature can be done after finishing single player part, as multi-player part is just a symmetric of single player

4. Discussion on game theme
   a. Game theme can be of commercial, educational or purely entertainment
   b. Educational game will require developers to think of what to educate
   c. Students suggested shooting as spiderman or iron man does
   d. Desmond suggested it can be a theme in the universe or space

5. Report on video demo for FYP
   a. It is not compulsory, but students think it is good to have video for game projects
   b. Will determine if do it or not based on schedule

6. Action Items
   a. Further discussion on game theme and design, should be more concrete
   b. Try out Leap Motion specific hand-gesture detection
   c. Try out spawning enemies

7. Meeting adjournment and next meeting
   a. The meeting was ended at 04:30 p.m.
   b. The next meeting will be on 16 August at 14:00 in Room 3521

Minutes of the 4th FYP meeting

Date: 16th August, 2016
Time: 2:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Sin Wing Lam

1. Demonstration on preliminary base project
    a. Base project for this demonstration includes hand control, shooting particle/ray and spawning enemy
    b. New headset - VR Box is introduced for more comfortable experience

2. Theme
    a. Basic emissive function in Unity is tested and cannot directly used in dynamic object
    b. Sin Wing Lam researched on some from Unity asset online
    c. A dark theme (e.g. Neon/ Space) and a bright theme (e.g. Forest) are chosen for testing the visual effect

3. 3D Model
    a. Dr. Desmond Tsoi suggested adding 3D Model into the preliminary project to test the capability of complex 3D Model

4. Extra items to add
    a. Dr. Desmond Tsoi suggested to have extra items in the list for future development after finishing the basic part
    b. Current ideas include:
        i. Tutorials
        ii. Mini-game
        iii. Social Media Components (e.g. Sharing)

5. Survey
    a. Dr. Desmond Tsoi suggested we can have survey asking users which theme will be better

6. Proposal
    a. Dr. Desmond Tsoi suggested to add some screencap into part 2 Methodology of the proposal

7. Action Items
    a. Update proposal for switching from Google Cardboard to VR Box
    b. Work on part 2 Methodology of the proposal
    c. Test import 3D model into preliminary project
    d. Test networking for multiplayer
    e. Buy a new headset

8. Meeting adjournment and next meeting
    a. The meeting was ended at 04:00 p.m.
    b. The next meeting will be on 25 August at 14:00 in Room 3521

Minutes of the 5th FYP meeting

Date: 25th August, 2016
Time: 1:30 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Demonstration of bright scene
    a. Sin Wing Lam showed scene with trees, grass and blue sky
    b. Desmond suggested investigating how we can adjust models' smoothness and quality. So the game can have balance between speed and graphics quality
    c. Should test the upper limit of our game rendering burden

2. Networking
    a. Tested using Photon Unity Networking (PUN) third party library
    b. It abstracts details like Socket, IP, etc. Using concepts like master server, lobby, room struct. Allowing an application to create room, join room and configure room options.
    c. Tried with bright scene, two players were able to see each other shooting bullets and the scene spawning enemies
    d. Performance is still acceptable

3. Proposal
    a. Updated Google Cardboard with VR Box
    b. Adding more pictures is encouraged
    c. General range of length of proposals is 30 to 40 pages
    d. For "Testing" section, can add test plan for each components. And can add integration test to test if anything crashed or mismatched
    e. For "literature survey" section, can add examples that uses VR, leap motion to support why our project is worthy to carry out
    f. For "Implementation" section, can have a bit technical descriptions, but no need to be too detailed

4. Additional features
    a. Walking
        i. Can consider if the game is able to detect walking motion
        ii. Johnny said had considered using hand gestures to control character movement
        iii. Desmond said using hand gesture is good because of safety, and can show that we have considered safety issues
    b. Multi games
        i. Johnny asked if multi game is preferred or single game is fine
        ii. Desmond suggested focus on one game and develop features surrounding that game, it shows that everything we developed are relevant and not digressed
    c. Sound
        i. Sound effects will be generated from PC
        ii. Consider outputting stereo sound based on user or camera position. For example, using Bluetooth speakers
        iii. Can test if unity API supports or not

5. Action Items
    a. Update proposal to include example of projects using VR technology and Leap Motion
    b. Update proposal to populated "Methodology" section
    c. Decorating "dark" scene
    d. Start implementing score and HP features

6. Meeting adjournment and next meeting
    a. The meeting was ended at 03:30 p.m.

b. The next meeting will be on 12 September at 19:00 in Room 3521

Minutes of the 6th FYP meeting

Date: 12th September, 2016
Time: 7:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Sin Wing Lam

1. Discussion on new network approach
    a. Originally bullets are network objects which each of their position, orientation and velocity are stored on the network. The new approach is to only send the spawn bullet message to a network manager (an RPC), and the message populates back to every client. And then bullets are spawn locally.
    b. No significant changes when playing the game, but it leaves more buffer in memory and performance for the future development.

2. Discussion on complexity test result
    a. Sin Wing Lam showed the complexity test result of the 3D models for the background scene
    b. A forest scene with more detailed is chosen to be the current theme
    c. Dr. Desmond Tsoi suggested the number of 3D models can be removed and more details can be added to the image for the skybox when the game cannot run smoothly

3. Proposal
    a. UAT can be added in the Part 2.3 Testing or Part 2.4 Evaluation
    b. Text alignment in Section “Division of Work” should be center aligned
    c. Every line should be aligned both left and right

4. Additional features
    a. Opening animation
        i. Dr. Desmond Tsoi suggested we can consider having an opening animation to appeal players

b. Transition between themes
    i. Dr. Desmond Tsoi suggested we can think of how the two themes we created can be linked together, are they consecutive levels with different level of difficulties, weapons or monsters. Or are they two different themes for player to choose which one they want.

5. Action Items
    a. Elaborate on Part 2.3 Testing and Part 2.4 Evaluation

6. Meeting adjournment and next meeting
    a. The meeting was ended at 08:30 p.m.
    b. The next meeting will be on 20 September at 19:00 in Room 3521

Minutes of the 7th FYP meeting

Date: 20th September, 2016
Time: 7:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Wong Chun Yin

1. Review of FYP proposal
    a. From another FYP team under Desmond, CT mentioned to add more technical terminology to the proposal.
    b. At the front page, FYP course code for this semester is COMP4981 instead of COMP4991
    c. Date of submission can input as 22nd September 2016
    d. Add and cancel out spacebar in appropriate places.
    e. Add back the page number
    f. Objective could use an introduction paragraph or rephrase the first time of each objective solution to show linkage between them.
    g. All pictures should be referred or mentioned in writings
    h. Replace the word “desktop” with computer / PC as we are running the game on a laptop in production
    i. Re-organize the sequence of the three requirements (Leap support, 3D rendering support, standalone game export)

j. Section 1.3.6 could use a picture
k. Section 2.1.2 Game Design can add a structure diagram of the project.
l. The existing section 2.1.2 can be moved to Testing
m. Rename Section 2.1.3, avoiding using the word “expected”
n. Section 2.1.3 also need a introductory paragraph the kick start the discussion
o. Current reason for choosing bright theme over dark theme is not strong enough. Need more elaboration.
p. Whole Section 2.2 Implementation should include more technical details. We may refer to the past FYP - Air Tennis.
q. Testing, Evaluation and Division of Work is fine.
r. We can make GANTT’s page landscape by adding page breaks
s. Items in GANTT chart can be a description line, depends on our team’s preference

2. Meeting adjournment and next meeting
a. The meeting was ended at 08:45 p.m.
b. The next meeting will be after proposal submission date in Room 3521, exact time and date will be further discussed

---

Minutes of the 8th FYP meeting

Date: 4th October, 2016
Time: 4:30 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Proposal
a. Proposal is already graded
b. Second reader is Professor Song(yqsong@cse.ust.hk), research interest in data extraction and machine learning

2. Monthly report
a. Monthly report can be submitted starting from now
b. Submission logistics

i. Writing the monthly report in Word document file
ii. Send the Word document to Dr. Desmond Tsoi
iii. Comment and sign by Dr. Desmond Tsoi
iv. Send back to student
v. Submit on FYP management system

c. September progress includes writing proposal, designing preliminary game scenes and testing software components(Unity, Leap Motion)

3. Implementation
   a. Two-weeks Goal will be set and review
   b. Wong Chun Yin and Lee Chun On will try to work on individual and separated tasks, but will still help each other if needed.
   c. Sin Wing Lam and Lee Chun On will try to work on related tasks

4. Action items
   a. Wong Chun Yin will work on scoring system
   b. Lee Chun On/Wong Chun Yin will work on energy HP
   c. Lee Chun On will work on music implementation
   d. Sin Wing Lam and Lee Chun On will work on UI display after the game started

5. Meeting adjournment and next meeting
   a. The meeting was ended at 05:30 p.m.
   b. The next meeting will be on 11 October at 16:30 in Room 3521

Minutes of the 9th FYP meeting

Date: 11th October, 2016
Time: 4:30 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Wong Chun Yin

1. Score System
   a. Attempt to use member Score in Player object, found that searching

player object after enemy destroy can be consuming, even with ObjectStore
   b. Discover PUN score extension, PhotonNetwork.player will always point to local PhotonPlayer object
   c. Score can be add and set in score extension, data will automatically sync to server

2. Scene Development
   a. Bright scene has been updated with better quality
   b. The background music using is from Lee Chun On's friend, there should not be a copyright issue

3. User Interface
   a. Problem encountered on Canvas due to split screen from Trinus. Currently have two approaches:
      i. Use script to split the canvas to two
      ii. Measure the coordinate and add canvas object to both camera

4. HP system and hit
   a. Two hit will kill an enemy
   b. Hit music is also implemented. It is played when bullet hit an enemy
   c. Hit will make enemy fall back

5. Meeting adjournment and next meeting
   a. The meeting was ended at 05:30 p.m.
   b. The next meeting will be on 18 October at 16:30 in Room 3521

Minutes of the 10th FYP meeting

Date: 18th October, 2016
Time: 4:30 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Wong Chun Yin

1. Progress Report

a. Updated UI
b. Health bar will adjust automatically when health changes
c. ScoreBoard will be used to wrap PhotonPlayer score related operation
d. Score on UI will get score from ScoreBoard

2. Short-term objectives
a. Change enemy model to something other than red cube
b. Enemy model available usually have skeleton attached
c. Implement observer pattern between ScoreBoard and Score on UI, avoid excessive update
d. Implement GetAllScore method in ScoreBoard to access score from all players, maybe using JSON

3. Meeting adjournment and next meeting
a. The meeting was ended at 05:30 p.m.
b. Due to career talk on 25 October, next meeting will be 1 November in Room 3521

Minutes of the 11th FYP meeting

Date: 20th December, 2016
Time: 2:30 p.m.
Place: Computer Barn A
Attending: Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: Dr. Desmond Tsoi
Recorder: Lee Chun On

1. New bright scene
a. Bright scene is updated due to slow performance
b. Bright scene is updated by Alice using assets from unity store: “Stylised Nature Pack” and “Cope! Free Skybox Pack”
c. Performance is better even using “Fantastic” mode in graphics settings

2. Tutorial scene
a. Tutorial scene is created by Jeffrey to teach users simple gesture control and gameplay control

b. Entry to tutorial scene through UI is not implemented, developer can enter tutorial scene by press “T” key in menu scene
c. Johnny said the instruction that guide users to show both hands is not clear enough
d. Tutorial scene will direct users to bright scene directly after completing

3. Reimplement weapon behavior and hit behavior
a. Weapon behavior and hit behavior are reimplemented by Johnny due to low scalability
b. Using strategy pattern to separate varying behaviors out of bullet classes for dynamic allocation
c. Two bullet types created: Bullet and knife
d. Three hit behaviors created: Normal, invulnerable, absorb

4. Discussion on creating new gesture for changing weapon
a. Flip left hand by 180 degree
i. Not applicable because Leap Motion often flips or misunderstand palm direction
b. Cross hands
i. Not applicable because Leap Motion is not accurate on two hands recognition when they overlap
c. Close both hands
i. Not applicable because some users may close their hands by default when playing games
d. Touching thumb with pointer/index finger
i. Leap Motion detection is relatively accurate
ii. Applicable and Jeffrey will try to implement this before next meeting

5. Taking video as a backup plan for presentation
a. The application occasionally not working when Wi-Fi network is not good enough
b. Decided to take gameplay video to demonstrate implemented feature as backup plan for future presentations
c. Need to investigate in what causing the error when Wi-Fi network is bad

6. Menu decoration
a. Alice will decorate the menu scene
b. Preliminary design is to use a new skybox and use prefabs that are used

in the game scenes to decorate

7. New levels
    a. Alice will create new levels by copying bright scene, follow by changing decorations and enemies

8. Ending a game
    a. The game will be ended when certain score is achieved, but is subject to change. For example, killing a boss or surviving from hugh wave of enemies
    b. End game when user blood level reaches zero
    c. Only display per game score in this stage
    d. Should be aware of multiplayer mode
        i. When one player die, the player will join as a bystander until the other die
        ii. When both players die, handle it with single player settings

9. Meeting adjournment and next meeting
    a. The meeting was ended at 03:15 p.m.
    b. The next meeting will be on 28 December at 10:30 in Room 3521

Minutes of the 12th FYP meeting

Date: 28th December, 2016
Time: 10:30 a.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Meeting Communication Tutor
    a. The meeting was re-scheduled to 11:00 a.m. on 5th January due to Johnny's personal reason
    b. Bring along the project proposal to Communication Tutor for reference
    c. Progress report will be due on 16th Feb, 2017

2. Tutorial scene

a. Tutorial scene was demoed, stages include:
    i. Displaying both hands
    ii. Leaving both hands
    iii. Shooting bullets with opened hand
    iv. Shooting enemies with bullets
    v. 10 seconds readying for the first stage
b. Tutorial scene was decorated by Alice
c. More instructions will be added in the future if needed, for example,
    i. Player can perform a thumb up gesture to skip tutorial

3. Redesign bright scene
    a. Bright scene was redesigned by Alice to resolve performance issue
    b. Bright scene was tested in production environment to ensure it meets the performance standard
    c. Skybox was used to mimic an extended view of the scene

4. End of game
    a. Score scene is showed when player health hit zero or press I (for testing)
    b. Player's name and score are displayed on a panel as UI, which moves along with camera
    c. Working on fade out or other effect indicating user's death
    d. Will implement end game on multi-player environment

5. New gesture detection
    a. New gestures will be used for changing weapon, skipping tutorial, etc.
    b. Leap Motion provides API for detecting circle, swipe, key tap and screen tap gestures (https://developer.leapmotion.com/documentation/v2/csharp/devguide/Leap_Gestures.html)
    c. Tested circle gesture, it is extremely sensitive that it will be triggered by moving a hand
    d. Tested swipe gesture, it is a bit too sensitive that it will be triggered when hand is moving in moderate speed
    e. Tested key tap gesture, it is not too sensitive and is best detecting a tap with the whole index finger
    f. Will try to add custom constraints after Leap Motion detection to detect more precisely

6. Meeting adjournment and next meeting

a. The meeting was ended at 11:45 a.m.
b. The next meeting will be on 10 January, 2017 at 14:00 in Room 3521

Minutes of the 13th FYP meeting

Date: 10th January, 2017
Time: 2:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Discussing feedback from Communication Tutor
    a. Should consider reordering and rearranging design and implementation sections into overview and objectives sections, but not rewriting everything
    b. Overview is better to contains abstract information, such as components used and simple software architecture. Descriptions should be informative but not too deep into details, details leave to design and implementation sections
    c. Objectives can be modified based on the modified Overview section, for example, add objectives for ShootingManager, EnemyManager and NetworkManager
    d. Literature Survey should still include background studies, but should consider combining or remove some similar sections, for example, maybe combining Leap Motion and VR camera into one section.
    e. Subtitles in design, implementation and testing sections should be reordered

2. Schedule for progress report
    a. Deadline is on 16th February, we target to have the first draft of progress report by 3rd February

3. Network issue in project
    a. Johnny discovered issue when working on multiplayer mode. If a player joins a game in the middle, blood information and bullet information of other player in player B view will be wrong

b. Alternative A is to start the game synchronously, every players start at the same moment
c. Alternative B is to request other players information as a payload when the player join the game

4. Hand detection issue in project
   a. Leap Motion unity API provides gesture detections such as swipe, circle, tap. However, they are not too accurate
   b. Jeffrey will try to combine default gesture detections and custom calculations to implement change weapon action first
   c. If the solution is not successful, alternative will be change weapons whenever player shoot some targets, for example, upgrade weapon whenever player shoots a gift bubble

5. Miscellaneous
   a. If time is limited, we should aim at completing the flow of the game instead of spending too much time on fancy features

6. Meeting adjournment and next meeting
   a. The meeting was ended at 3:30 p.m.
   b. The next meeting will be on 18 January, 2017 at 14:00 in Room 3521

Minutes of the 14th FYP meeting

Date: 18th January, 2017
Time: 2:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Sin Wing Lam, Lee Chun On
Absent: Wong Chun Yin
Recorder: Lee Chun On

1. Demonstration of software progress
   a. Network issue discovered in multiplayer mode is resolved by using OnPhotonSerializeView function provided by PUN
   b. Change weapon gesture detection problem is resolved by detecting a

shaking gesture of a closed-hand
c. New weapons are demonstrated, such as, fire rock, ice wave, lightning fissure
d. Space scene is redecorated to introduce new enemies, such as, alien and astronaut
e. A new scene, MapScene, is developed as a portal to select different stages
f. For the video, Jeffrey have tested using BandiCam for recording avi and converting the avi to ogg on the internet followed by importing to Unity. However, the graphics quality drops when converting to ogg, need investigation

2. Presentation issue
   a. For FYP presentation, students prefer to present features with the help of pre-recorded video. And after the presentation, a portion of time will be reserved to demonstrate the game or invite supervisor to try
   b. Dr. Desmond Tsoi said there is no restriction on how the presentation should be organized, but the demonstration is important and prefer to have it

3. Project enhancement
   a. Dr. Desmond Tsoi suggested that we can collect and analyse player's shooting data and give some performance statistics and suggestions for improvements
   b. Jeffrey suggested that the game can capture a few pictures for player to download or share after the game

4. Miscellaneous
   a. Before next meeting, the main focus will be writing the first draft of progress report

5. Meeting adjournment and next meeting
   a. The meeting was ended at 3:10 p.m.
   b. The next meeting will be on 2 February, 2017 at 14:00 in Room 3521

Minutes of the 15th FYP meeting

Date: 2nd February, 2017
78

Time: 4:30 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Lee Chun On

1. Progress report discussion
    a. Introduction section
        i. Objective 2 is shortened and details of selecting suitable VR camera are moved to implementation
        ii. Game engine literature survey is removed
    b. Methodology section
        i. Design and implementation are redesigned to include 4 sections, they are “Game Flow”, “Model Design”, “Hardware Design” and “Software Architecture”
        ii. Design part is mainly brief descriptions and figures for readers to have basic overview
        iii. Implementation part contains great details of how the designs are implemented, what decisions we have made, what we have done so far and what will be done in the future
    c. GANTT
        i. Dr. Desmond Tsoi reminded to make sure to include all the implemented tasks in the progress report, do not hide or miss any details
    d. First draft should be submitted to Dr. Desmond Tsoi by 3rd February

2. Game performance issue
    a. Noticeable flashing occurs on mobile phone every 10 to 20 seconds
    b. No flashing occurs on PC
    c. Johnny suspected that is due to Trinus version update
    d. Dr. Desmond Tsoi suggested that we revert back to an older version of our game or Trinus to see if the problem still exists

3. Meeting adjournment and next meeting
    a. The meeting was ended at 5:30 p.m.
    b. The next meeting will in Room 3521, exact time and date will be further discussed

---

Minutes of the 16th FYP meeting

Date: 13th February, 2017
Time: 5:00 p.m.
Place: Room 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Wong Chun Yin

1. Progress report discussion
    a. Due to busy schedule, Desmond will review the progress report tonight (13/02) and give out comments by tomorrow
    b. The screen caps in the progress report leave a positive impression and demonstrate a good progress of the project

2. Project-related sharing
    a. Jeffrey shares a Doraemon project in Japan which use Leap Motion with HTC Vive to simulate Dokodemo Door.
    b. Can be used in introduction but avoid very details intro, which leads to audience's comparison between that project and ours.
    c. Desmond shares Professor Pan Hui's AR project on CSE website

3. Additional idea
    a. Boss fight
    b. Special wave of enemy if player performs good
    c. May includes element that relates to future job hunting (e.g. video streaming)

4. Meeting adjournment and next meeting
    a. The meeting was ended at 6:30 p.m.
    b. Due to Progress Report submission, regular meeting on coming Thursday is postponed.
    c. The next meeting will in Room 3521, exact time and date will be further discussed

---

Minutes of the 17th FYP meeting

Date: 23rd March, 2017
Time: 3:30 p.m.
Place: Barn A
Attending: Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: Dr. Desmond Tsoi
Recorder: Lee Chun On

1. Bug report
    a. In pull request #47, score list decoration, horror scene is still showing bright scene background. Jeffrey will be fixing this.
    b. For a better user experience, ask player to extend their hands in the first stage of tutorial. Jeffrey will handle this in issue 50.
    c. Leap Motion hand model bugs, Johnny will handle this in issue #49
        i. when about to change scene, keep your hand within leap detection (hand model appear in the scene)
        ii. in the new scene when you try to move the hand out of leap detection, hand model will stay at the position of the last position. (hand number can go infinite)

2. New features
    a. A progress bar on the top of the screen to show how many enemies remaining and the boss blood level. Alice will be implementing this.
    b. Replacing player with a proper prefab
    c. Disable collisions between players

3. Testing of the equipments
    a. FYP presentation will be on April 29 (Saturday) at 9:50-10:30am at Room 2127B.
    b. Jeffrey proposed that our team should test the whole set of equipments in Room 2127B before presentation
    c. Currently decided that Johnny and Jeffrey will test it on March 29 (Thursday) after meeting.

4. Meeting adjournment and next meeting
    a. The meeting was ended at 4:30 p.m.
    b. The next meeting will be on 30th March, 2017 at 2:00p.m. in Room 3521

Minutes of the 18th FYP meeting

Date: 30th March, 2017
Time: 4:30 p.m.
Place: Rm 3521
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Wong Chun Yin

1. Presentation arrangement
    a. FYP presentation will be on 29/4 in Rm 2127b
    b. The second group to present in that day
    c. Can arrive early (maybe 8:30am) to set up the environment
    d. 40 min presentation includes the setup time, prefer direct connection via cable regarding scene sharing
    e. Prepare some questions that will be asked (e.g. Any interviewer asked about your FYP? What is your view on VR application in the future? What additional feature you would like to add if you have time (big data)? Leap Motion application on other field?)
    f. The team is going to test the setup after this meeting in Rm2127b
    g. Desmond suggest to book a room for full demo practice, inviting some audiences

2. Final report
    a. Meeting with communication tutor tmr (31/3)
    b. Desmond suggest to finalize the project by 10/4 and start working on the report.

3. Progress
    a. Jeffrey mentioned two updates
        i. Tutorial add a part to show players how to change bullet
        ii. Map integration to whole game flow
    b. Ideas
        i. Menu in scene
        ii. Progress bar that shows number of remaining minions and boss health

4. Meeting adjournment and next meeting
    a. The meeting was ended at 6:00 p.m.
    b. The next meeting will be on 6th April, 2017 at 4:30p.m. in Room 3521

Minutes of the 19th FYP meeting

Date: 6th April, 2017
Time: 3:30 p.m.
Place: Barn A
Attending: Wong Chun Yin, Sin Wing Lam, Lee Chun On, Dr. Desmond Tsoi
Recorder: Wong Chun Yin

1. Status report
    a. Desmond mentioned that he confirmed other FYP teams agree to have presentation practice together. Desmond will book a venue for that when we confirm the attendance and time.
    b. Jeffrey demo the whole game flow to Desmond, from menu scene to score scene, and back to menu. During the demo, we notice a repeating bug occurs again, a fix is planned.

2. Improvement
    a. Use the scenes we have and change them into different difficulty level. Add an option to play next level once he finish the game
    b. Add mechanism to allow player get buff, for example having shield generated in front of player to block enemy from approaching
    c. Revamp the score scene to have better visibility of information
    d. Add 10 seconds countdown at the end of tutorial scene for better indication of how long should the player wait
    e. May add environmental effect such as snowing, leaf falling

4. Meeting adjournment and next meeting
    a. The meeting was ended at 4:30 p.m.
    b. The next meeting will be on 13th April, 2017 at 4:30p.m. in Room 3521

Minutes of the 20th FYP meeting

Date: 13th April, 2017
Time: 4:30 p.m.
Place: Rm 3530
Attending: Dr. Desmond Tsoi, Wong Chun Yin, Sin Wing Lam, Lee Chun On
Absent: None
Recorder: Sin Wing Lam

1. Project Progress Demo
    a. We had a demo of the whole project
    b. The flow is completed and music & sound effect matched with the environment
    c. Can add more sound effect (e.g. different sound for different weapons, sound of enemy walking) and make the background music softer to emphasize more on the sound effect

2. Final report
    a. Desmond suggested to add more screenshot for each part to give a better overview
    b. For the Discussion section, we should include challenges or limitation that we faced, contribution of the project to the related field (i.e. Game and VR industry), possible improvement for future work.
    c. Also, we can put the justification of decision that we made into Discussion section
    d. For the Conclusion section, we should include the summary of the project, and what we have done and achieved, but this should be short and about 1-1.5 page
    e. Desmond also suggested to elaborate more technical details and we can include the design pattern that we used

3. Presentation
    a. During the presentation, we should show what we have considered and designed throughout the whole process.
    b. We prefer to have a complete rehearsal before the actual presentation, but depends on time
    c. If the marker wants to try, we prefer to let him try and it gives him a better understanding of the project

d. If there are any crashes during demonstration, Desmond said it will be fine, and we should try to recover quickly

4. Poster
    a. More screenshots should be included to illustrate our game better and easier to attention
    b. The major content in the poster should be our idea, with brief description of technology used
        i. Also, be focus on the main achievement Connecting Leap Motion with VR, and VR game with networking feature.
        ii. If there are still some spaces, more technical details can be included, such as enemy spawning and collision detection
    c. Link can be added as QR code for reader's reference
    d. 21/4 is the deadline to submit the poster in ppt format for printout

5. Meeting adjournment and next meeting
    a. The meeting was ended at 6:00 p.m.
    b. The next meeting will be on 20th April, 2017 at 4:30p.m. in Room 3521