

DEVELOPMENT OF INTERACTIVITY GAME  
USING FACIAL LANDMARKS

BY  
GOH THENG KHAI

A PROPOSAL  
SUBMITTED TO  
UNIVERSITI TUNKU ABDUL RAHMAN  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
BACHELOR OF INFORMATION SYSTEMS (HONS)  
INFORMATION SYSTEMS ENGINEERING

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY  
(PERAK CAMPUS)

MAY 2017



**UNIVERSITI TUNKU ABDUL RAHMAN**  
**FACULTY OF INFORMATION & COMMUNICATION**  
**TECHNOLOGY (PERAK CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	14ACB00092
Student Name	Goh Theng Khai
Supervisor Name	Dr. Ng Hui Fuang

<b>TICK (✓)</b>	<b>DOCUMENT</b>
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

\_\_\_\_\_  
(Signature of Student)  
Date:

\_\_\_\_\_  
(Signature of Supervisor)  
Date:

## REPORT STATUS DECLARATION FORM

**Title:** Development of Interactivity Game using Facial Landmarks

**Academic Session:** May 2017

I GOH THENG KHAI

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

\_\_\_\_\_  
(Author's signature)

\_\_\_\_\_  
(Supervisor's signature)

**Address:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
Supervisor's name

**Date:** \_\_\_\_\_

**Date:** \_\_\_\_\_

DEVELOPMENT OF INTERACTIVITY GAME  
USING FACIAL LANDMARKS

BY  
GOH THENG KHAI

A PROPOSAL  
SUBMITTED TO  
UNIVERSITI TUNKU ABDUL RAHMAN  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
BACHELOR OF INFORMATION SYSTEMS (HONS)  
INFORMATION SYSTEMS ENGINEERING

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY  
(PERAK CAMPUS)

MAY 2017

## **DECLARATION OF ORIGINALITY**

I declare that this report entitled “DEVELOPMENT OF INTERACTIVITY GAME USING FACIAL LANDMARKS” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENTS**

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Ng Hui Fuang who has given me this bright opportunity to engage in a game project. It is my first step to establish a career in game design and development field. A million thanks to you.

To a very special person in my life, Abigail, for her patience, unconditional support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

## **ABSTRACT**

Nowadays, there are millions of games in the market but only a few of them do not require physical input while some of them are expensive as they require specific hardware requirements to execute. This project is to develop an interactive motion-control game using facial landmarks detection technique for low-end and mid-range Android mobile devices. Basically, the game should be able to detect and track the real-time user facial landmarks movement and face expression to perform some interactions between user and device. Hence, it will provide user a novelty interactive gameplay experience unlike the traditional way of using touchscreen and device coordination to control the game. Moreover, several related applications, including FaceRig, MSQRD and SnapChat, have been reviewed and their strengths and weaknesses have been discussed. To have better implementation on facial landmarks detection, some face recognition and detection techniques, including Google Mobile Vision Face API, Microsoft Cognitive Service Face API and Luxand Face SDK have been reviewed for figuring out their features, implementation methods and platform compatible issues. Moreover, agile development process is selected and come out with some appropriate prototypes, logic flows, storing methods, development tools, requirements and timelines for project visualization. Implementation and system verification are conducted before releasing the finalized game. The concept of project can be extended to other applications in future.

## Table of Contents

DECLARATION OF ORIGINALITY .....	i
ACKNOWLEDGEMENTS .....	ii
ABSTRACT .....	iii
Table of Contents .....	iv
List of Figures .....	vii
List of Tables .....	xi
List of Abbreviations .....	xii
Chapter 1: Introduction .....	1
1.1 Project Background .....	1
1.2 Problem Statement and Motivation .....	3
1.3 Scope .....	4
1.4 Objective .....	4
1.5 Impact, Significance and Contribution .....	4
1.6 Innovation .....	5
1.7 Proposed approach/study .....	6
1.8 Report Organization .....	8
Chapter 2: Literature Review .....	9
2.1 Review Description .....	9
2.2 Review on Existing Games and Applications .....	9
2.2.1 FaceRig .....	9
2.2.2 MSQRD .....	12
2.2.3 SnapChat .....	14
2.3 Comparison between Applications using Facial Landmarks Techniques .....	16
2.3.1 Features Explanation .....	17
2.4 Critical Remarks of previous works .....	18
2.5 Summary .....	20



2.6 Reviews on Face Recognition and Detection Techniques .....	21
2.6.1 Google Mobile Vision Face API .....	21
2.6.2 Microsoft Cognitive Service Face API.....	23
2.6.3 Luxand Face SDK .....	24
2.7 Comparison between Face Recognition and Detection Techniques .....	26
Chapter 3: System Design.....	27
3.1 Top-Down System Diagram .....	27
3.2 Functional Block Diagram .....	28
3.3 Use Case Diagram.....	29
3.4 Activity Diagram.....	31
3.5 UML Diagram .....	39
3.6 Class Diagram .....	41
3.7 ERD Diagram.....	50
3.8 Low-Fidelity Game Prototypes .....	52
Chapter 4: Design Specifications and Implementations .....	55
4.1 Methodology .....	55
4.2 Software and Hardware Requirements.....	56
4.2.1 Software Requirements.....	56
4.2.2 Hardware Requirements .....	58
4.3 User requirements .....	59
4.4 System Performance Definition .....	60
4.5 General Work Procedure .....	62
4.5.1 Planning .....	62
4.5.2 Analysis .....	62
4.5.3 Design.....	62
4.5.4 Implementation and Testing (Coding and Development) .....	63
4.6 Project Timeline .....	64

4.7 Detailed Implementation .....	66
4.7.1 Step-by-step to compile the program.....	66
4.7.2 Communication between Android Java and Unity C# Backend .....	71
4.7.3 Game Scenes Concept and Description.....	72
4.7.4 Data Storing Technique – Shared Preferences .....	73
4.7.5 High-Fidelity Game Prototypes .....	74
4.8 Verification Plan and Program Testing .....	84
Chapter 5: Conclusion.....	88
5.1 Project Achievements.....	88
5.2 Project Reviews.....	88
5.2.1 Project Strengths.....	88
5.2.2 Project Limitations .....	88
5.3 Implementation Issues and Challenges .....	89
5.4 Future Work .....	91
5.5 Summary .....	92
Bibliography .....	93

## List of Figures

Figure 1-1: Facial Landmarks and Face Recognition Preview .....	2
Figure 1-2: Overall game system flowchart .....	6
Figure 2-1: FaceRig PC gameplay (1) - Face changed based on facial landmarks .....	9
Figure 2-2: FaceRig PC gameplay (2) - Parts of face customization .....	9
Figure 2-3: FaceRig mobile phone gameplay (3) - Recognized face before playing ...	10
Figure 2-4: Masquerade application preview on Android platform (1) .....	12
Figure 2-5: Masquerade application preview on Android Platform (2) .....	12
Figure 2-6: SnapChat application preview on Android Platform (1) .....	14
Figure 2-7: SnapChat application preview on Android Platform (2) .....	14
Figure 2-8: The face orientation detection used by Mobile Vision Face API .....	21
Figure 2-9: The Facial “landmark” detection .....	21
Figure 3-1: Top-Down System Diagram .....	27
Figure 3-2: Block diagram of the game .....	28
Figure 3-3: Upper half of use case diagram .....	29
Figure 3-4: Remaining half of use case diagram .....	30
Figure 3-5: Game System (Main) Activity Diagram .....	31
Figure 3-6: High Score Module Activity Diagram .....	32
Figure 3-7: Camera Activator Module Activity Diagram .....	33
Figure 3-8: Item Store Module Activity Diagram .....	34
Figure 3-9: Game Configuration Module .....	35
Figure 3-10: Game Development Details Module .....	36

Figure 3-11: Game Starter Module .....	37
Figure 3-12: Upper half of UML diagram for Face API's backend (Java) .....	39
Figure 3-13: Remaining half of UML diagram for Face API's backend (Java) .....	40
Figure 3-14: Simpler Tweeners Class Diagram .....	41
Figure 3-15: Sound Manager and UV Scroller Class Diagram .....	42
Figure 3-16: Shop Item Class Diagram .....	43
Figure 3-17: Power-Ups Class Diagram .....	44
Figure 3-18: Menus Class Diagram .....	45
Figure 3-19: Level Generations Class Diagram .....	46
Figure 3-20: Height Motion, Slide Motion and Random Scaler Class Diagram .....	47
Figure 3-21: Game Manager, Player Control and Scene Selector Class Diagram .....	48
Figure 3-22: Spaceship Battle Class Diagram .....	49
Figure 3-23: Face Controller Class Diagram .....	50
Figure 3-24: Entity-relationship diagram (ERD) .....	51
Figure 3-25: Main Interface .....	52
Figure 3-26: Item Purchased Interface .....	52
Figure 3-27: Single Player Interface .....	53
Figure 3-28: Multi-Player Interface .....	53
Figure 3-29: Settings Interface .....	54
Figure 3-30: Gameplay Interface .....	54
Figure 4-1: Incremental Development Model .....	55
Figure 4-2: Black-Box Testing Model .....	63
Figure 4-3: The Upper Half of the Project Gantt Chart .....	64

Figure 4-4: The Remaining Half of the Project Gantt Chart .....	65
Figure 4-5: Unity Game Development Engine New Project Initialization Page .....	66
Figure 4-6: Unity Game Development Engine Project Selection Page .....	66
Figure 4-7: Unity Game Development Engine Main Interface .....	67
Figure 4-8: Unity Game Development Engine Build Settings .....	67
Figure 4-9: Android Studio Interface – Manifest Configuration .....	68
Figure 4-10: Android Studio Main Interface .....	69
Figure 4-11: Developer option in Android device .....	69
Figure 4-12: Android Studio Program Deployment Interface .....	70
Figure 4-13: Android Studio Finalized Project Deployment Interface .....	70
Figure 4-14: Visualized Communication between Java and C# languages .....	71
Figure 4-15: Storing Method – Shared Preferences (XML) .....	73
Figure 4-16: Main Menu Interface .....	74
Figure 4-17: Game Option Selection 1 Interface .....	74
Figure 4-18: Game Option Selection 2 Interface .....	75
Figure 4-19: Game Instruction Gameplay Interface .....	75
Figure 4-20: Gameplay 1 Interface .....	76
Figure 4-21: Pause Menu Interface .....	76
Figure 4-22: Restart Menu 1 Interface .....	77
Figure 4-23: Restart Menu 2 Interface .....	77
Figure 4-24: Gameplay 2 Interface .....	78
Figure 4-25: Item Store Interface 1 .....	78
Figure 4-26: Item Store Interface 2 .....	79

Figure 4-27: High Score Board Interface .....	79
Figure 4-28: Game Settings Interface .....	80
Figure 4-29: Game About Interface .....	80
Figure 4-30: Testing Mode Interface 1 .....	81
Figure 4-31: Testing Mode Interface 2 .....	81
Figure 4-32: Testing Mode Interface 3 .....	82
Figure 4-33: Enlarged Camera Interface .....	82
Figure 4-34: Program Exit Interface .....	83

## List of Tables

Table 2-1: Comparison between existing games and applications .....	16
Table 2-2: Features Explanation .....	17
Table 2.3: Strengths and Weaknesses of FaceRig .....	18
Table 2.4: Strengths and Weaknesses of MSQRD .....	18
Table 2.5: Strengths and Weaknesses of Snapchat .....	19
Table 2.6: Strengths and Weaknesses of proposed game .....	20
Table 2.7: Comparison between Face Recognition and Detection Techniques .....	26
Table 4-1: User Stories .....	59
Table 4-2: Non-functional requirement .....	61
Table 4-3: Game Launch Verification Plan .....	84
Table 4-4: Game Main Menu Verification Plan .....	84
Table 4-5: Game Market Store Verification Plan .....	85
Table 4-6: Single Player Configuration Verification Plan .....	85
Table 4-7: Multi-player Configuration Verification Plan .....	86
Table 4-8: Game Configuration Verification Plan .....	86
Table 4-9: Gameplay Verification Plan .....	87

### **List of Abbreviations**

<i>2D</i>	Two-Dimension
<i>3D</i>	Three-Dimension
<i>API</i>	Application Program Interface
<i>APK</i>	Android Application Package
<i>AR</i>	Augmented Reality
<i>CPU</i>	Central Processing Unit
<i>ERD</i>	Entity-Relationship Diagram
<i>FPS</i>	Frame Per Second
<i>GB</i>	Gigabyte
<i>GHz</i>	Gigahertz
<i>GPS</i>	Global Positioning System
<i>GPU</i>	Graphics Processing Unit
<i>GUI</i>	Graphical User Interface
<i>IDE</i>	Integrated Development Environment
<i>Inc</i>	Incorporated
<i>JNI</i>	Java Native Interface
<i>JSON</i>	JavaScript Object Notation
<i>NFC</i>	Near Field Communication
<i>RAM</i>	Random Access Memory
<i>ROM</i>	Read-Only Memory
<i>SDK</i>	Software Development Kit
<i>UML</i>	Unified Modelling Language



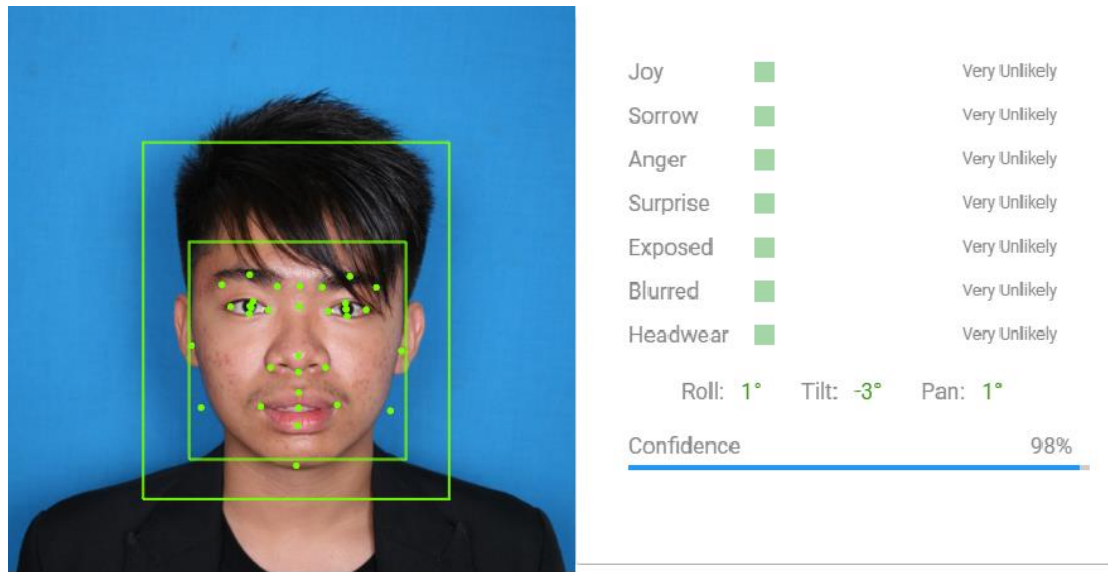
## **Chapter 1: Introduction**

### **1.1 Project Background**

Face recognition is a computer technique that has ability to identify or recognize a person based on the characteristics of his or her face. Normally, a digital image or a video frame from a video that contains human face will be used to analyse and extract its meaningful features and characteristics in terms of inner features and outer features such as eyes, nose, mouth, head shape and hairline. After that, comparing and matching them with a facial database then putting them into a useful representation and performing some kind of classification on them.

Basically, face recognition is considered quite an easy task for a human being to identify known faces. Nevertheless, human being has limited memory and not skilful on dealing with a large amount of unknown faces. Major computers in the existing current market have quite good specifications where they able to solve this limitation of human efficiency and effectively.

Besides, face recognition technique grows maturely and becomes popular as rapid evolution of the technology. For instance, there are a lot of existing image processing libraries and products in current market and able to detect human face accurately and automatically. As camera and computer hardware have been popularized, a high-quality camera and a better specification of hardware like central processing unit (CPU), graphic processing unit (GPU), random access memory (RAM) disk, read-only memory (ROM) disk, storage drive can be obtained easily since its cost has been reduced significantly compared to 20th century. Nowadays, this technique plays an important role in human life and implemented in many areas including information security, access management, biometrics, law enforcement, personal security and entertainment.



*Figure 1-1: Facial Landmarks and Face Recognition Preview (Google Cloud, 2016)*

Facial landmarks detection plays an important role on initializing most of the common face analysis jobs where these jobs consist of recognizing, verifying and inferencing human face and some related attributes. According to figure 1-1, the face recognition engine required a digital image with human face to detect the face information in terms of lines, edges, angles, landmarks and movement then display the information of human face based on the computer-based visual cortex system and database.

In this project, an interactive game using facial landmarks is selected and it will be developed for certain platform. The term interactive means that user can interact the game by using his or her face as input through the camera unlike the traditional way of using hand to control the game. Basically, facial detection technique is the main project concern but not facial recognition where the game will only locate where the users' faces but not to recognize who they are so it is not necessary to have a facial database and it can probably work offline. Furthermore, facial classification technique can also be included since the game may require some face activities like winking, blinking and smiling as some additional inputs.

## 1.2 Problem Statement and Motivation

First of all, majority of existing games in current market are lacking of human interaction. For instance, keyboard, mouse and touch screen are used as input devices on personal computers (PC), laptops or mobile phones where devices detect the changes of physical key or coordinate the user input gather on device screen only. Those input devices may be still worked effectively but users might be getting bored of using them to play games.

Secondly, general motion-control games depend on specific hardware or adapters, including Microsoft Kinect for Xbox, PlayStation Move and Wii, in order to perform and play where they are very expensive. In the meantime, majority of the related high-end games are produced and released by well-known firm so the cost will be much higher. As a result, there is only a certain group of consumers are able to afford, test and experience it. Thus, it is unfair to those consumers whose have less income but willing to own it.

Thirdly, a webcam or build-in camera is a must and required to run simultaneously with motion-control games so memory resources and power resources consumption will be getting larger. This is a significant issue if the games run on laptop or mobile device since these devices come with pre-built hardware specifications and do not allow to change or update. As a result, these devices may have a better chance of getting overheating and cannot last longer due to large battery consumption. If these devices either do not have sufficiency of processing power or failed to meet the minimum system requirements, then the frame per second (FPS) of the game graphic and performance of controlling game will be degraded significantly. Thus, user gameplay experience will be harmed a lot.

Fourthly, there is a few segments of people who are worked in indoor and have to stay in front of the PC for long periods will have less face and head movement. In long term, these people may have face muscle laxity where it may affect their health and appearance.

### **1.3 Scope**

This project is to develop an interactive game using facial landmarks on Android mobile platforms. Camera is required to perform detection and tracking of facial landmarks movement to interact with the game content. Basically, this game is targeted to all users with low-end and mid-range devices because it is only required lesser processing power and system requirements where the devices are easily to be obtained with a cheaper price.

### **1.4 Objective**

To develop an interactive game using facial landmarks as user can interact with the game content with his or her facial landmarks movement through camera, unlike the traditional way of using hand to control the game.

1. To detect and track real-time facial landmarks movement through camera.
  - As long as the camera is activated, the game should be able to detect and track the user facial landmarks movement smoothly with the target of minimizing all the possibilities of detecting facial landmarks latency issues.
2. To control game operations based on facial landmarks movement.
  - The game should record the patterns of facial recognition from time to time.
  - Game operation like changing the direction of game object should only be involved using facial landmarks recognition so that user can play the game without touching the device for a specific distance.
  - Once the facial recognition patterns meet a specific requirement, the game should trigger some special events on screen.

### **1.5 Impact, Significance and Contribution**

This project is based on facial landmarks for controlling the game where it may prevent users of getting bored like traditional game playing method using hands and fingers. Basically, it fully depends on camera as all the user inputs can only obtain from video frame. Without camera lens, the game could not be function. Fortunately, nowadays cameras are everywhere as mobile phones and laptops come with pre-built camera or user with PC can afford an external camera such as webcam with a low price. Camera

is used to detect and track the face landmarks movement in order to control and interact with the game. Initially, the game will require user to recognize his or her face to ensure current user has the only privilege to control the game as this configuration can avoid others' faces from accidentally control the game if camera captured more than a person. Then, the game will work with the camera to trace motion of face movement to interact with the game contents.

Moreover, this project will focus on developing low and medium graphic motion-control game where it has been utilized for lower phone system requirements and less battery consumption so that it can be run smoothly on most of the devices included low-end devices. Thus, low income users are able to experience it without purchasing other motion-control games since low and mid-range devices can be purchased in an ideal cost. It also can make users close to the technology of face recognition.

Furthermore, the way of increasing face movement is to play this proposed game. With the attractiveness of user interface and game content, users will automatically adapt to the playing method by moving their heads and faces.

### **1.6 Innovation**

In this project, the interactive game using facial landmarks with camera to interact with the game content. In the proposed solution, it is more concentrated on the facial landmarks detection instead of game graphic design.

Besides, this proposed game should have the option of having single and multi-player mode. For multi-player mode, it is required to connect with the technology of Bluetooth, NFC and Wi-Fi direct in order to interact with other users.

In addition, this game should have online scoring board so that every user is able to interact with other users by completing the scores respectively. Basically, user is required to connect to another web services like Google Plus or Facebook in order to activate this service.

Furthermore, this game should provide a market store where user is able to change to some appearance of game objects. Also, the store can provide some additional game modes that are required certain amount of coins to unlock where the coins can be acquired through game.

## 1.7 Proposed approach/study

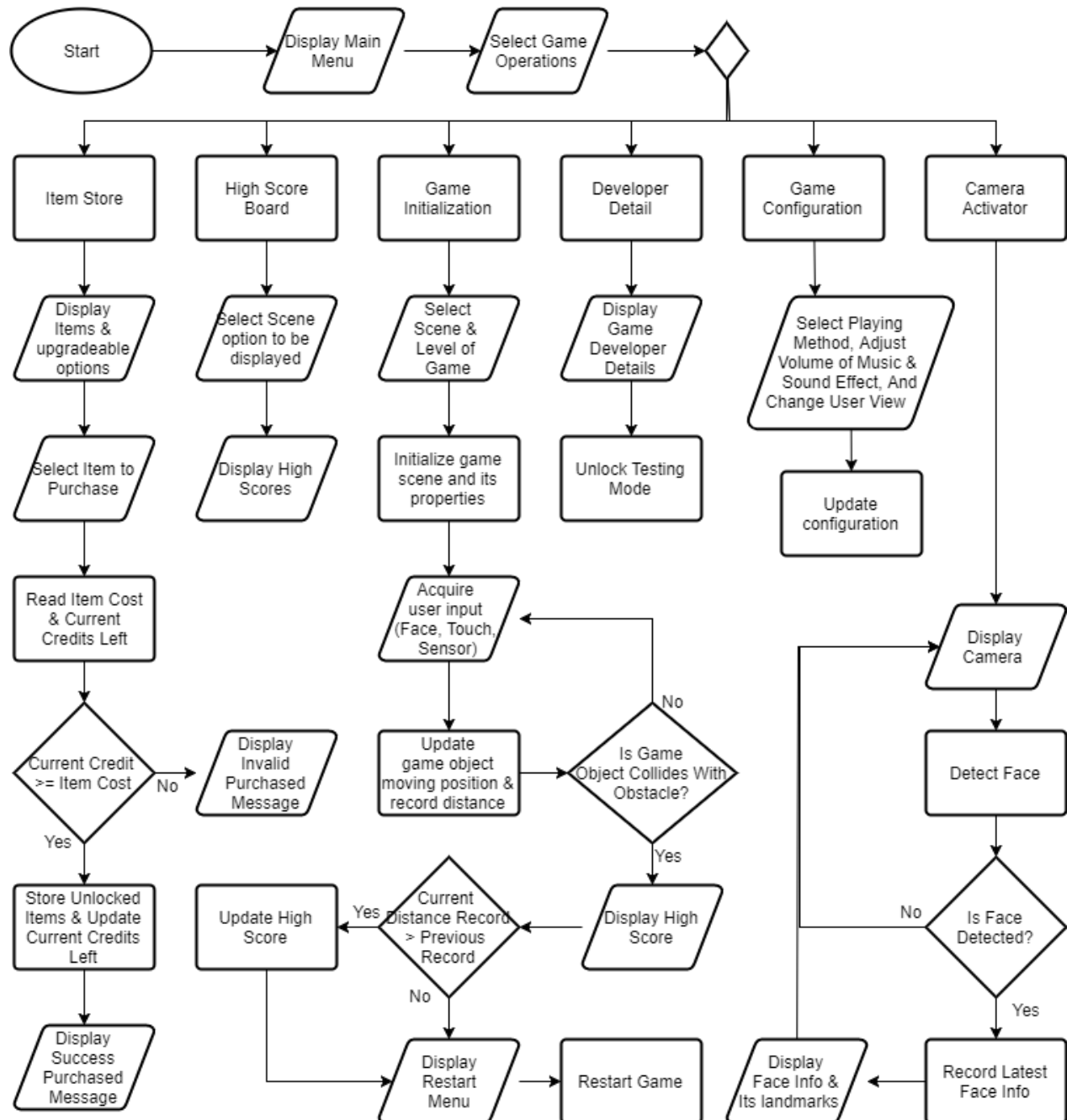


Figure 1-2: Overall game system flowchart

Basically, the game system consists of 6 major modules where the modules included game initializer, camera activator attached with facial landmarks detection, item store, high score board, game configuration and game development detail attached with testing mode. Firstly, game initializer module is used to construct all the game objects

in game scene based on game settings and user selection, control and manipulate the game objects based on user input and calculate high score and credits obtained. Secondly, camera activator module is used to detect facial landmarks from users' face, read facial expression and calculate the movement values for x-axis and y-axis before send the message back to the game system. Thirdly, item store module is used to handle all the item purchased transaction and game object customizations based on items availability, items' costs and current amount of credits that users obtained. Fourthly, high score board module is used to display various high scores based on different playing method and game scene. Fifthly, game configuration module is used to make some changes on game settings based on user own preferences. Lastly, game development detail module is just used to display developer details but it contains a hidden testing mode which includes advance game settings. It mainly used for testing purpose otherwise users will be lost fun from playing the game. Thus, in order to come out with these modules, there are some techniques which are required to learn such as manage and manipulate 2D and 3D graphic objects and implement face API for retrieving facial values.

## 1.8 Report Organization

The report is totally based on final year project's guideline which updated on 3<sup>rd</sup> July 2017 by Universiti Tunku Abdul Rahman (UTAR), it included appropriate contents, formats and alignments.

Basically, the report divides into 5 chapters and each chapter contains its own unique contents while interrelating with each other. Firstly, chapter 1 is mainly focused on project introduction and planning phrase. It contains project background, problem statements, project objectives and scopes, impact, significant, and contribution of project, project innovations, proposed study and the overall project progress.

Secondly, chapter 2 is concentrated on analysis phrase. It contains some literature reviews about a few of existing games and applications that have implemented similar techniques and some face detection APIs where they can reduce project workload. Each review will be compared and contrasted for obtaining their pros and cons.

Thirdly, chapter 3 is primarily described system designs and flows. Thus, it contains various types of diagrams, including top-down system diagram, block diagram, use-case diagram, activity diagram, UML diagram, class diagram, ERD diagram and expected game prototypes for better project visualization.

Fourthly, chapter 4 includes the combination of all the development phrases. These phrases contain methodology, development tools, user and system requirements, general work procedures, project timeline, detailed implementations in term of program compilation, program logic and flow description, storing method and the actual program's appearances, and verification plan that used to compare with the behaviours of real program.

Lastly, chapter 5 is used to make conclusion about the overall project. It contains implementation issues, project achievements, program pros and cons, potential improvements for future, and project summary.



## Chapter 2: Literature Review

### 2.1 Review Description

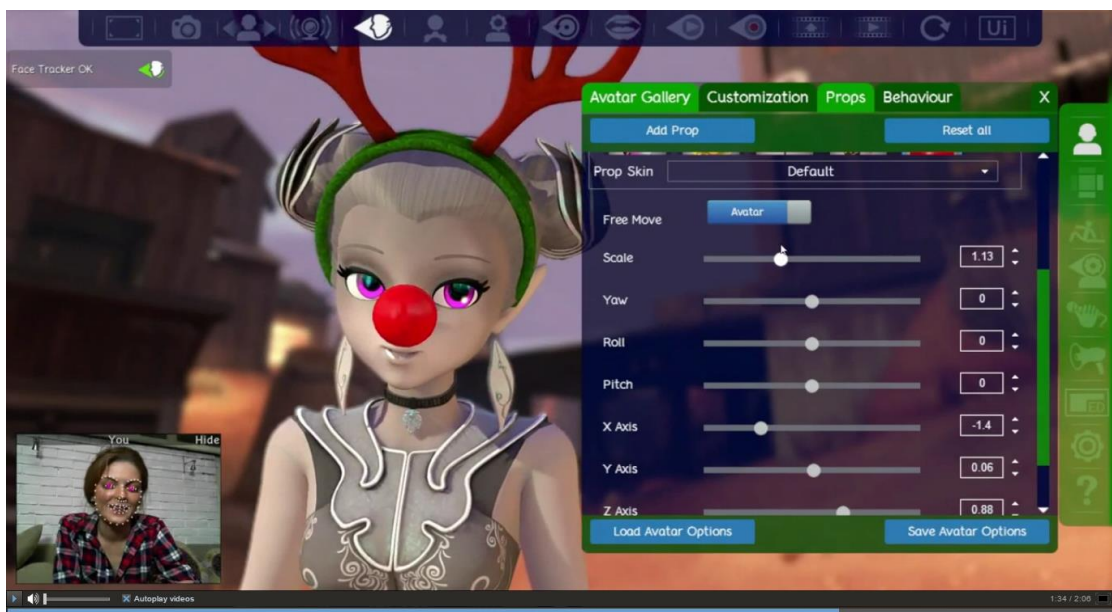
There are a few examples of existing games using facial landmarks technique that has been reviewed as the following.

### 2.2 Review on Existing Games and Applications

#### 2.2.1 FaceRig



*Figure 2-1: FaceRig PC gameplay (1) - Face changed based on facial landmarks*



*Figure 2-2: FaceRig PC gameplay (2) - Parts of face customization*



*Figure 2-3: FaceRig mobile phone gameplay (3) - Recognized face before playing*

FaceRig games was developed by Holotech Studios and its purpose was to use a webcam or a camera for embodying users into awesome characters in real time where the characters will change according to users' face expressions and emotions so it could categorize as an augmented reality (AR) game. It was available on both PC and mobile platforms, including Windows OS, Android OS and iOS, where its PC version was required to pay with the minima of RM31.00 while its advance version cost RM115.00 whereas its mobile platform was free of charge. (FaceRig, 2015).

In PC version, it supported screenshot, video recording and streaming features where it could work concurrently with any real-time service using webcam such as Skype, Twitch and Hangouts. On the other hand, mobile phone version like Android did not support streaming features. (Indiegogo, 2014).

In order to play the game, user must let the game to detect user's face within a certain range between user and camera where it was just like figure 2-3. If the range was too big or too small, the game will unable to recognize user face. Fortunately, it still could proceed further to embody user's face but its stability and chances of identifying user's face will be reduced significantly. Besides, user could select various face virtually painting options, including avatars, live2D, masks and current background before another operation like streaming will be performed. Then, the output will be

transformed according to the facial landmarks of current user. If certain conditions had been triggered such as opening mouth, then an additional effect will be included in the output as figure 2-3 shown. Once video recording operation had been finished, then it will provide some options of sharing to any social media applications directly within the applications. By default, there was quite a few number of avatars could be used while the rest of them were required to unlock where they could be unlocked with a certain amount of coins which could be obtained through daily login and in-app purchase. The level of user will be gained after certain avatars success to unlock. Hence, it might motivate and attract users to play. (Holotech Studios, 2016).

Furthermore, there were a few notables from this game. For PC version, when the game was on video recording mode, users might not able to hear their own voice due to the loud hissing and static noise. Although it could be slightly solved by installing another software which was called LoiLo but all the voice configurations will not be saved after a restart from the program. (Clatworthy, 2016).

For mobile platform like Android, the accuracy and performance of detecting users' faces slightly worse and slow when it compared with PC version. When another avatar was selected, it required about 5 seconds to process whereas PC version could be switched immediately. Besides, the speed of the game will be degraded after run for a certain period and it might cause the mobile devices to have black screen issue where the devices force to reboot for overcoming it. Thus, the optimization of the game for mobile version was encouraged to be improved since those issues were serious and annoying to users where they could bring down the reputation and popularity of the game. Last but not least, there were only a limited number of free faces and avatars available for users where the rest of them were required to pay via in-app purchasing option.



### 2.2.2 MSQRD

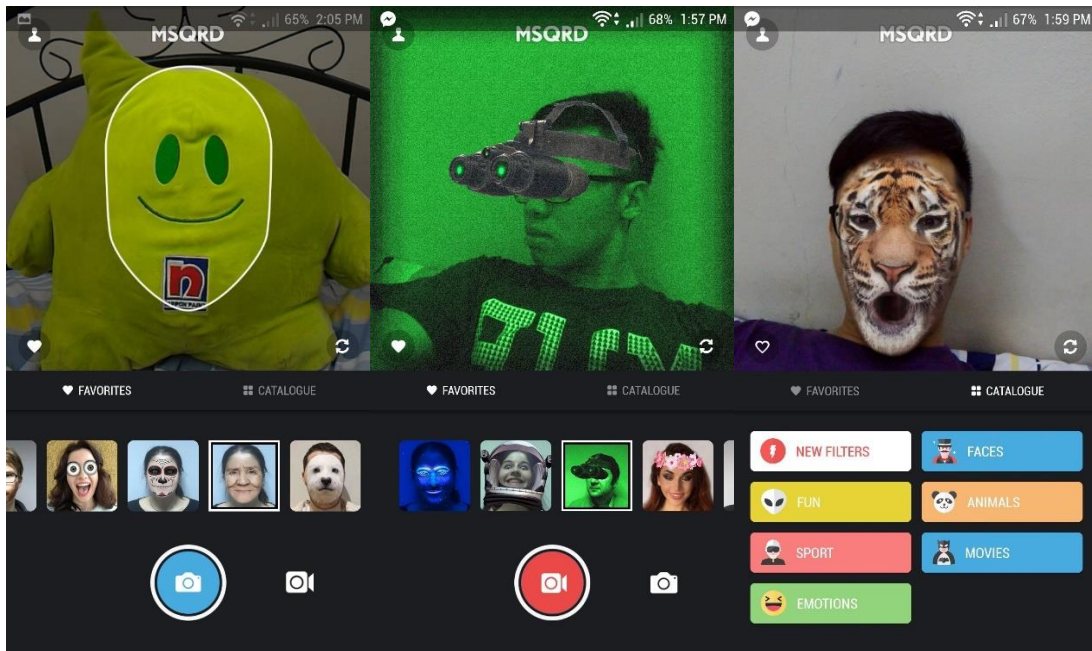


Figure 2-4: Masquerade application preview on Android platform (1)

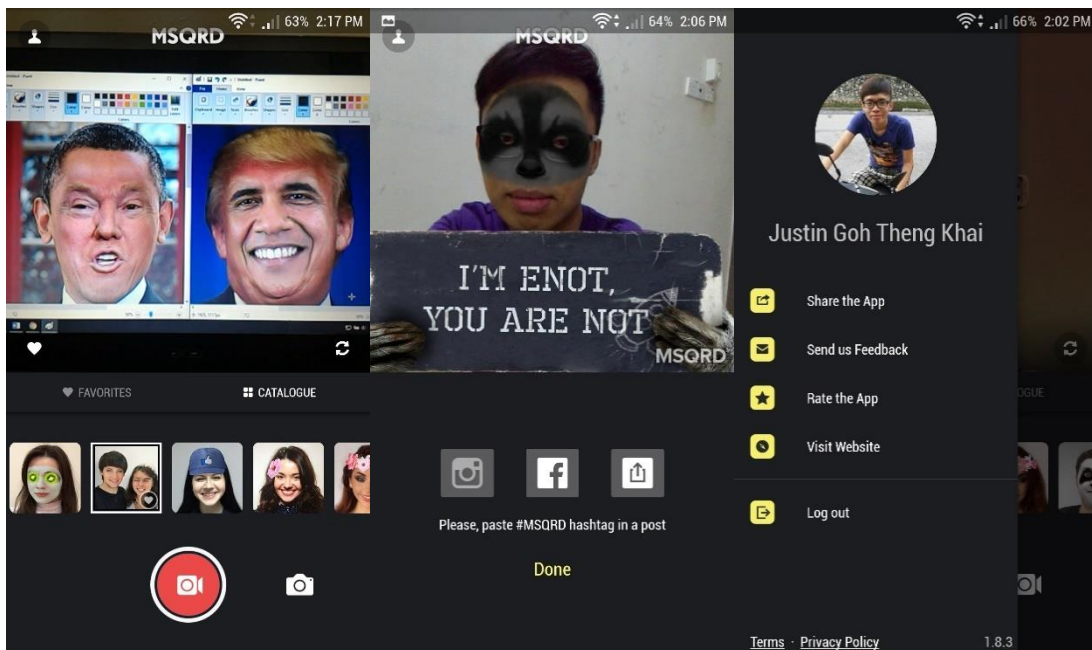


Figure 2-5: Masquerade application preview on Android Platform (2)

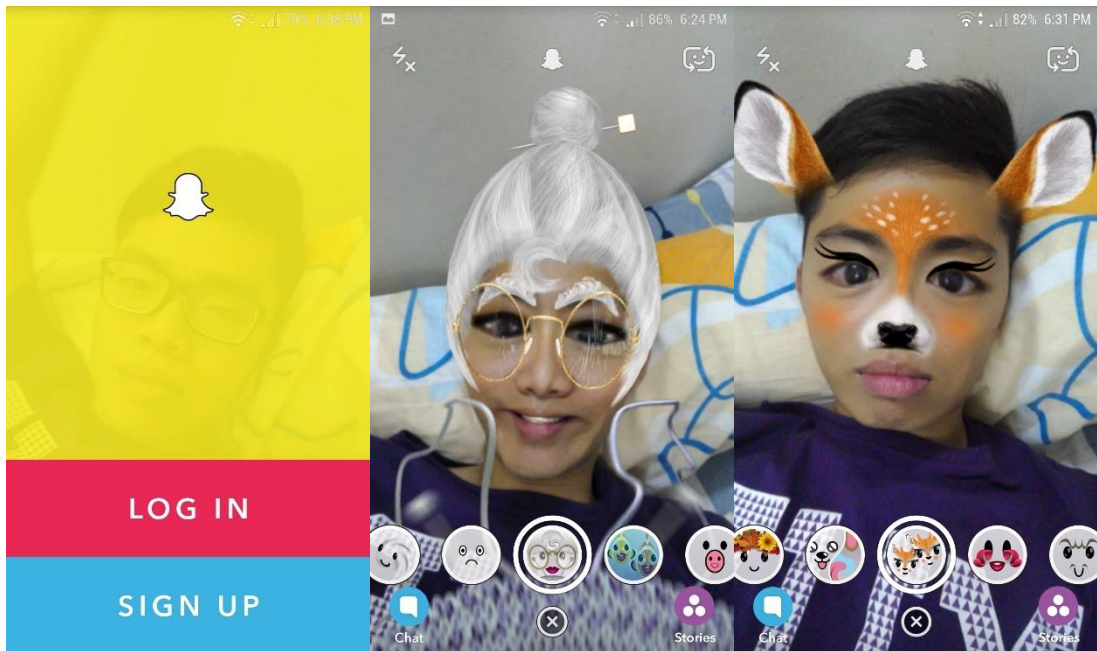
MSQRD was another mobile application using the technology of facial expression recognition, 3D face tracking and AR for changing the face appearance of user in a real-time camera where it developed by Masquerade Technologies and acquired by Facebook Inc. (Smith, 2016).

Basically, once the application had been launched, it will display social media login page where users could choose to either login or ignore it. Then, the front camera will be activated by default along with a series of masks in different catalogues so that it was convenient to users from selecting it since it provided quite a number of masks which might take some time of figuring out it. If the device did not contain front camera, then back camera will be switched. If users' faces were unable to recognize, then the application will display a human head shape to notify that users' faces were not in the camera screen. If users' faces had been identified successfully then the info screen will be disappeared immediately as it shown in figure 2-4. After that, users could enjoy of playing it by taking pictures and recording videos. There is an option to share the outputs directly to social medias or other connectivity once pictures and videos had been finished capturing. (Facebook, 2016).

This application required the basic permissions, including take pictures and videos, record audio, storage access and full network access from users' devices according to the purpose and objective of using it. Besides, it did not have any advertisement so that users will not be annoyed from it. Also, every single mask was freely to be selected and used since it was totally free without any in app purchasing option. The application installer size was considered regular since basic masks were only included while the additional masks needed to be downloaded within the application via online. Also, the application was stable to use and run smoothly since it consumed quite less processing time and RAM. Thus, users should be happy, confident and having fun from using it regardless of the users' ages.

On the other hand, according to the reviews from Google Play, it seemed like some of users were unhappy with this application on Android platform since the features and availability of masks were quite lesser compared to iOS platform where it seems to be unfair to Android users. (Facebook, 2016). Based on the situation, the developers were encouraged to update their application for multi-platform simultaneously. Furthermore, the captured pictures and videos force to store in local storage without providing any option to remove it. It considered one of the inconvenient matters where users required to delete the items using other applications. It was encouraged to add an option to ask for the permission of saving the pictures and videos into local storage instead of directly storing them.

### 2.2.3 SnapChat



*Figure 2-6: SnapChat application preview on Android Platform (1)*



*Figure 2-7: SnapChat application preview on Android Platform (2)*

Note: This review will only focus on image processing, face recognition and facial landmarks movement detection while the rest of the features was being likely to be ignored since they were not related to this project.



Snapchat was a messaging mobile application developed by Snap Inc. since year 2011, where it was similar to WhatsApp, WeChat and Messenger but it contained its own unique features included sharing text, images and videos to a list of friends or worldwide for a specific period and they will be self-destructed from the recipients and server once it reached the deadline, and customizing users' faces on real-time camera that supported up to 24 FPS based on users' own preferences using the technique of AR, face expression recognition and facial landmarks detection. (Cabebe & Mitroff, 2013).

Basically, when user entered the application, front camera will be activated as it shown in figure 2-6. When a user launched the application for the first time, the application will display a yellow start-up screen for the configuration of login and sign up. User needed to have an account in order to proceed further. Once configuration has been set up, the information will be stored in the device so user did not need to login for the following time. In order to activate the facial customization option, user was required to face to front camera within a certain range and tapped on his or her own face on the device screen then the application will start to recognize the face pattern. This process will be taken around couple of seconds based on the device's specifications. After that, a list of custom faces will be displayed and user could select the his or her favourite face. These faces had their special effects respectively and they could enable various kind of beauty facial effects to custom and theme users' faces. If user met certain selected custom face's conditions like opening mouth, shaking head and moving eyebrow, then some extra effects will be triggered and added into the screen. Besides, the application could detect the faces from user's local image storage and then grabbed them to override current user's face. While playing with any single effect, user could choose to capture images and videos then either shared them to other friends or stored in local storage. (Snap, 2016).

The overall features of the application, including user interface, functionality, performance and stability, were quite good. Nevertheless, some Android devices were incompatible with it. (Snap, 2016). Furthermore, the issue of forcing users to login or sign up using SnapChat account for proceeding further operations was not flexible at all. It suggested to provide some additional login options such as Facebook and Google Plus where new user could register an account in a more convenient way.

### 2.3 Comparison between Applications using Facial Landmarks Techniques

Game & Application Features	FaceRig	MSQRD	SnapChat	Proposed Game
Camera	✗	✗	✗	✗
Flashlight			✗	
Microphone	✗	✗	✗	
Vibration Control	✗		✗	
Bluetooth			✗	✗
Internet Access	✗	✗	✗	
GPS			✗	
Manage Storage Rights	✗	✗	✗	
Account Required		Optional	✗	
Prevent Sleeping	✗		✗	✗
Hardware Requirement	High	Low	Moderate	Low
Power Consumption	High	Moderate	Moderate	Moderate
AR Technique	✗	✗	✗	
Able to pause the application				✗
Finger interaction on screen within application	✗	✗	✗	Configuration before start the game
Number of users (X) for each session in a device	1	$1 \leq X \leq 2$	$1 \leq X \leq 2$	1

Table 2-1: Comparison between existing games and applications



### 2.3.1 Features Explanation

Table 2-2 consists of the explanations of applications' permissions and features.

<b>Features</b>	<b>Explanation</b>
<b>Camera</b>	To capture pictures and videos.
<b>Flashlight</b>	To lighten the environment when camera is activated.
<b>Microphone</b>	To record voice and sound.
<b>Vibration Control</b>	To notify user from specific events.
<b>Bluetooth</b>	To access Bluetooth settings and pair devices.
<b>Internet Access</b>	To access internet for items downloading purpose.
<b>GPS</b>	To locate the location of devices.
<b>Manage Storage Rights</b>	To have rights to manage device local storage.
<b>Account Required</b>	Required login or sign-up before proceed further.
<b>Prevent Sleeping</b>	To avoid devices from getting into sleeping mode.
<b>Hardware Requirement</b>	The specification of devices like processor and chipset
<b>Power Consumption</b>	The power usage of applications.
<b>AR Technique</b>	Application Involves in augmented reality.
<b>Able to pause the application</b>	To pause the application state when it is running.
<b>Finger interaction on screen within application</b>	The need of tapping devices' screens when using the application.
<b>Number of users for each session in a device</b>	The number of users to interact with the application using a single device.

*Table 2-2: Features Explanation*

## 2.4 Critical Remarks of previous works

There are three existing mobile games have been reviewed from the previous records. First and foremost, FaceRig is a face detection game using AR technology that embodies users into awesome characters in real time. The face of user will be transformed according to certain face expressions. It also offers virtually painting options, including avatars, live2D, masks and background. User can have selfie and video recording when playing it. It offers varieties of high-quality graphic components that are quite attractive but the rendering graphic process for mobile version may not be as smooth as PC version.

Strengths	Weaknesses
<ul style="list-style-type: none"> <li>• Offer variety of face customizations</li> <li>• Does not require user account to login or sign up</li> </ul>	<ul style="list-style-type: none"> <li>• In-app purchased option for certain features</li> <li>• High hardware requirement</li> <li>• High power consumption</li> <li>• Unstable performance for certain devices</li> <li>• Advertisement included</li> </ul>

*Table 2.3: Strengths and Weaknesses of FaceRig*

Secondly, MSQRD is another facial expression recognizing mobile application using 3D face tracking and AR technique for transforming the user's face appearance. This application required the least hardware permissions, including take pictures and videos, record audio, storage access and full network access from users' devices according to the purpose and objective of using it. It is no longer have update since August 2, 2016 as most of the features have been integrated to Facebook and Messenger.

Strengths	Weaknesses
<ul style="list-style-type: none"> <li>• Light-weighted installer package</li> <li>• Free without advertisement</li> <li>• Low hardware requirement</li> <li>• Does not require to have account</li> </ul>	<ul style="list-style-type: none"> <li>• Imbalance features between Android and iOS platform</li> <li>• Stop having update on Android platform</li> </ul>

*Table 2.4: Strengths and Weaknesses of MSQRD*

Last but not least, Snapchat is a messaging mobile application which included face expression recognition and facial landmarks detection techniques. It offers lists of custom faces with special effects like beauty facial effects to user. Besides, it the application could detect the faces from user's local image storage and then grabbed them to override current user's face. This is one of the special features where it makes the application to be more interesting and successfully attract user to try on it.

Strengths	Weaknesses
<ul style="list-style-type: none"><li>• Variety of attractive custom face features</li><li>• Recognize face from gallery and adopt them to customize face</li><li>• Good performance on most of the devices</li><li>• Average hardware requirement</li><li>• Average battery consumption</li><li>• Free without advertisement</li></ul>	<ul style="list-style-type: none"><li>• Incompatible issue for certain devices</li><li>• User account required</li><li>• Inflexible account login method</li></ul>

*Table 2.5: Strengths and Weaknesses of Snapchat*

## 2.5 Summary

From the above reviews, all of them have their own pros and cons. All the reviews are related with face detection and AR technology so they have too many similarities. FaceRig for PC version is such a strongest among all where it provides lots of face customizations with the support of online video call whereas it may be some weaknesses on mobile version. As mobile devices come with limited hardware specification, it might affect the game performance due to high graphic requirement needed.

The proposed game using facial landmarks detection technique is considered a brand-new idea. There are couples of applications using this technology but none of them developed in game platform. Basically, the game provides a novelty gameplay experience as it only required the movement and expression of user's face to control and trigger some events while the game graphic will not be focused too much. The goal of the game is to move further by avoiding obstacles within certain time and certain playable chances. Therefore, front camera is a must in to play this game. Although back camera can be used, user is unable to view and play the game. Fortunately, most of the mobile devices provide a better quality of front camera. As the result, majority of users are able to try it without any compatible issues. The strengths and weaknesses of the proposed game are as below.

Strengths	Weaknesses
<ul style="list-style-type: none"> <li>• Offer new gameplay experience</li> <li>• Low hardware requirement</li> <li>• Less battery consumption</li> <li>• Play the game without touching the device</li> <li>• Free without advertisement</li> <li>• Multi-player mode support</li> <li>• Offer variety of customizing features.</li> <li>• Global ranking and high score viewer</li> </ul>	<ul style="list-style-type: none"> <li>• Require a short distance between user and mobile device</li> <li>• Optimal light intensity environment is preferable</li> <li>• Front camera is required</li> </ul>

*Table 2.6: Strengths and Weaknesses of proposed game*

## 2.6 Reviews on Face Recognition and Detection Techniques

### 2.6.1 Google Mobile Vision Face API

The mobile vision face API was developed by Google Inc where it provided face detection functionalities that allowed the usage of detected face features as input to trigger certain actions in the application. Currently, the API supports for face detection but not for face recognition in the mobile application development in both iOS and Android platform.

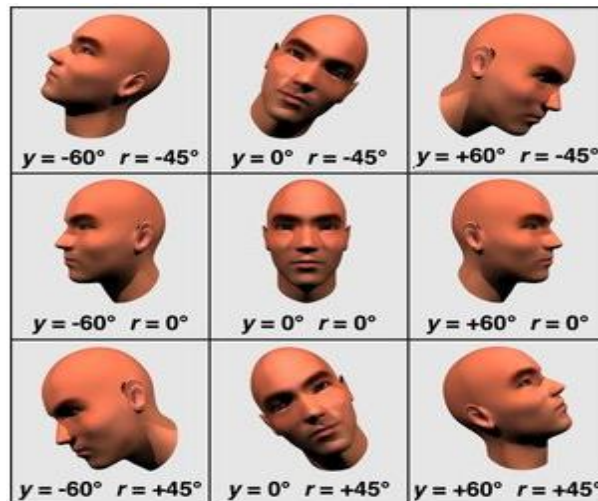


Figure 2-8: The face orientation detection used by Mobile Vision Face API



Figure 2-9: The Facial “landmark” detection

Besides, mobile vision face API detected human face using several approaches such as face orientation (Figure 2-8), facial landmarks (Figure 2-9) and facial classifications. Face orientation used the human face angles as estimation units to detect their face position and coordinate even if the human face was not front-facing the camera. Optionally, with facial landmarks, the API able to detect faces through different points on human face such as left-right eyes, nose, cheek, mouth and ear position known as landmarks. Furthermore, facial classification able to perform recognition of facial characteristics such as smiling and eyes open. It required the application of prior landmark detection results and confident interval value calculation in order to present the precise classification.

Moreover, mobile vision face API is free to use and supports for both Android and Apple development platforms. The API is still under development and potential for future improvement such as additional features add-on that give the API the ability to recognize facial features and categories based on facial traits. The mobile vision API unlike its cloud-based counterpart, the cloud vision API, had the benefit to implement offline rather than require network connectivity.

In the nutshell, since the API required integrated Google service framework to function properly, the devices without the framework cannot run the application with Google vision APIs built-in. Secondly, the facial landmarks detection method will require additional time to compile and run. Therefore, the developers specify the face orientation as default detection method and facial landmarks detection as optional method. (Google Mobile Vision, 2015).

### **2.6.2 Microsoft Cognitive Service Face API**

The face API was developed by Microsoft Corporate and it was a part of its Microsoft cognitive service APIs collection. The API emphasized on detecting and working with face data. It was often used in conjunction with other APIs to develop high-standard and machine-learning based application.

Besides, the face API offered functionalities which provide the application the abilities to deal with various face detection and recognition works. It can detect human faces on images, videos as well as in real-time environment through its complex algorithms that able to recognize face characteristics and return it in the form of JSON format. The result can be incorporated into other Microsoft cognitive service APIs to perform sophisticated tasks such as machine learning and depth analysis.

Furthermore, the face API can recognize and compare the face of individuals in distinct images in order to determine the probability of how similar an individual can be in another photo based on his or her face features. For instance, the face API from Facebook able to identify and search for individuals in a single photo based on their facial features.

In addition, with face API, image search is attainable by analysing the person faces against the provided model data. The model data (images) with close resemblance will be returned to the user. Furthermore, grouped and organized unfamiliar faces into similar category based on shared facial traits is another remarkable feature of face API which allows the application to classify people accordingly.

Last but not least, the face API comes at different pricing options. The free option allows user to consume the face API up to 30,000 transactions per month and 20 transactions per minute. Additional cost will be charged upon exceeding the limit which introduced user to the standard pricing option that allows user to pay differently for certain transaction range. Microsoft also provides the face storage service for user who wants to store relevant images as a database collection for the usage in application development. (Microsoft Cognitive Service, 2016).

### 2.6.3 Luxand Face SDK

The face SDK was developed by Luxand Inc where it is a powerful development kit which had been used widely in the application development that required a lot of image and video processing works. It supports variety of vision libraries which provide the developer the flexibility to build advance and powerful application using human faces data as main inputs. It supports for various development platforms ranging from OS, web to mobile development.

First and foremost, the face SDK supports for various features which are related to computer vision processing. Firstly, the ability to recognize and identified human faces with its unique motion-based face recognition techniques. This reliable technique able to recognize and tag individuals appear in the real-life videos. The tagged individuals can be personalized, registered and stored in face database which can be used in future identification work rather than reanalysing and recognize again for the specific individuals. Secondly, with the built-in algorithms to recognize and detect individuals' similarities, the SDK able to successfully identify individuals who appeared in images and videos by looking up for existing face database records.

Secondly, the face SDK able to detect human faces present in the images and return their relevant facial landmarks coordinates. This feature utilized algorithms which can detect human facial features with more than 70 landmark points to indicate every facial feature position precisely (Luxand, 2016). Additionally, this feature can use in real-time videos where every human face in the video frames will be detected and marked for counting purpose.

Thirdly, other than typical face detection and recognition, the face SDK supports for special detection and recognition such as eye movement tracking, facial expression or emotion tracking and gender detection. All these special features are applicable in images as well as videos. It provides more options to developer for developing application which can react to user eye moment, emotions and gender attributes.

Fourthly, image manipulation is included in development kit to enable various image manipulation functionalities such as image cropping, resizing, reassembling and others to be used as part of the application development. Besides, the thread management feature allows developers to fully utilize processors to support for multi-concurrency activities. This can optimize the performance of the application when dealing with huge



volume of data processing. Moreover, the face SDK can be integrated into compatible cameras and webcams that support for direct video streaming so that data can be retrieved for analysis usage.

Lastly, being the powerful development kit for computer vision development, the face SDK has few notable points to be mentioned. Basically, it supports wide-range of development platforms which include mobile platform such as Android and iOS. Next, it continues to improve their existing structure in order to simplify the integration process with other projects. A complete documentation is available for developer in order to guide them through how to use the development kit effectively and efficiently. Finally, in order to ensure no frustrating experience among developers, the face SDK team continues to test and improve so that developer can focus on development and deployment without concern about potential bugs and security issues. (Luxand, 2017).

## 2.7 Comparison between Face Recognition and Detection Techniques

Face Techniques Features	Google Mobile Vision Face API	Microsoft Cognitive Service Face API	Luxand Face SDK
Work offline	✗		Optional
Framework dependency	✗	✗	✗
Documentation provided	✗	✗	✗
Free of charge	✗		
Internet required		✗	Optional
Face recognition support		✗	✗
Facial landmarks detection support	✗	✗	✗
Facial classification support	✗	✗	✗
Face database involved		✗	✗
Android and iOS platform compatibility	✗	✗	✗
Number of facial landmarks detected per human face	Up to 12 points	Up to 27 points	Up to 70 points
Library/installer package size	Tiny	Small	Large

*Table 2.7: Comparison between Face Recognition and Detection Techniques*

## Chapter 3: System Design

### 3.1 Top-Down System Diagram

Figure 3-1 shows that the total expected number of program interfaces which interlink respectively and it gives numerous ideas to proceed further project development. The overall detailed flows can refer to figure 3-5.

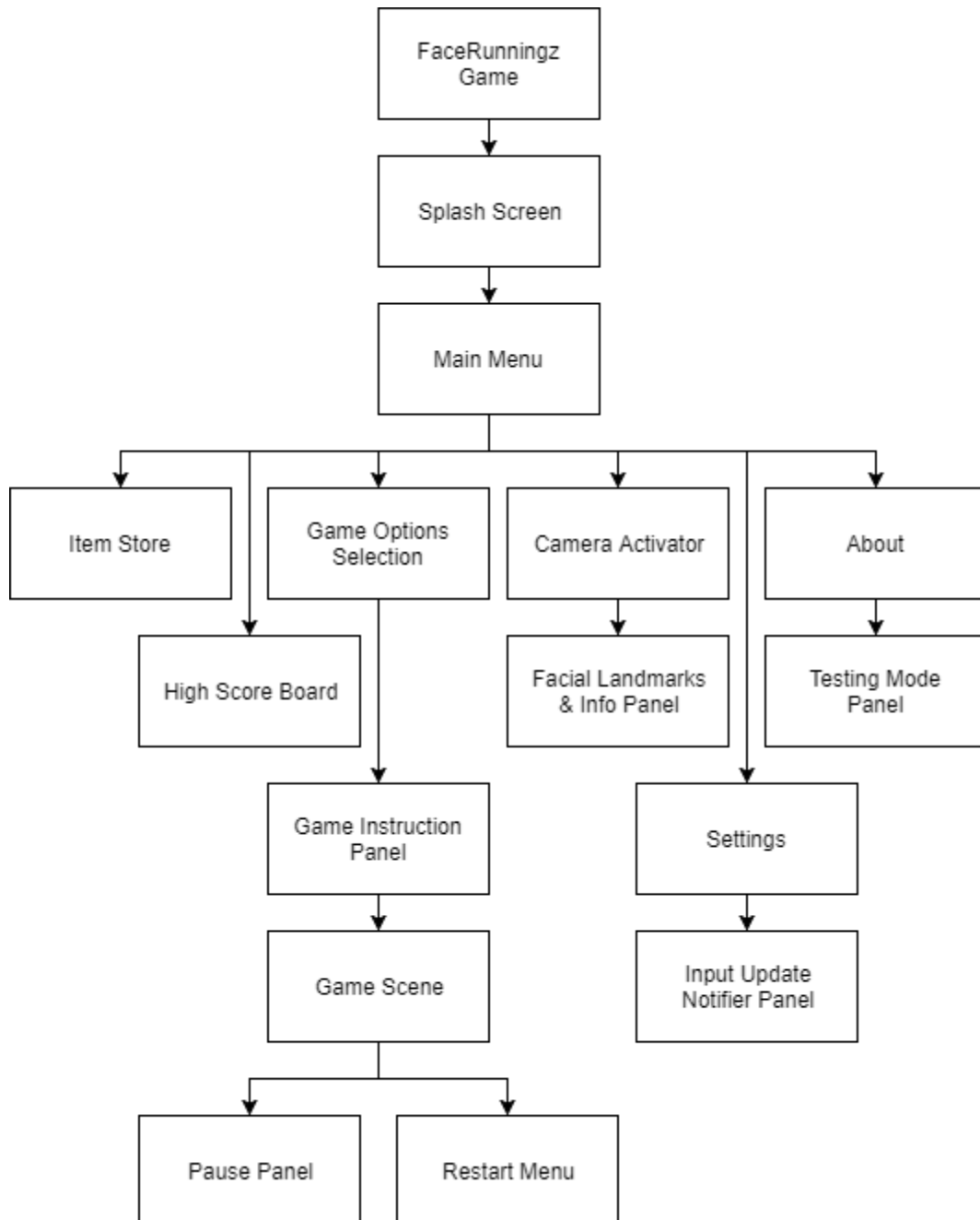


Figure 3-1: Top-Down System Diagram

### 3.2 Functional Block Diagram



Figure 3-2: Block diagram of the game

#### Description:

- **Facial Landmarks Detector & Face Expression Recognizer:** User can play the game using front camera by determining his or her facial landmarks and face expression within a distance without touching his or her device.
- **High Score & Leader board Viewer:** User can view his or her high score locally and rank globally with some integrated social media application services such as Facebook and Google Plus.
- **Item Store:** User can activate some special modes and abilities for better playing experience.
- **Game Customization:** User can customize his or her appearance of game controller object with a variety of options from item store.
- **Game Customization:** User can change game configurations based on his or her preferences.
- **Testing Mode:** Developer can unlock testing mode for debugging purpose.

## 3.3 Use Case Diagram

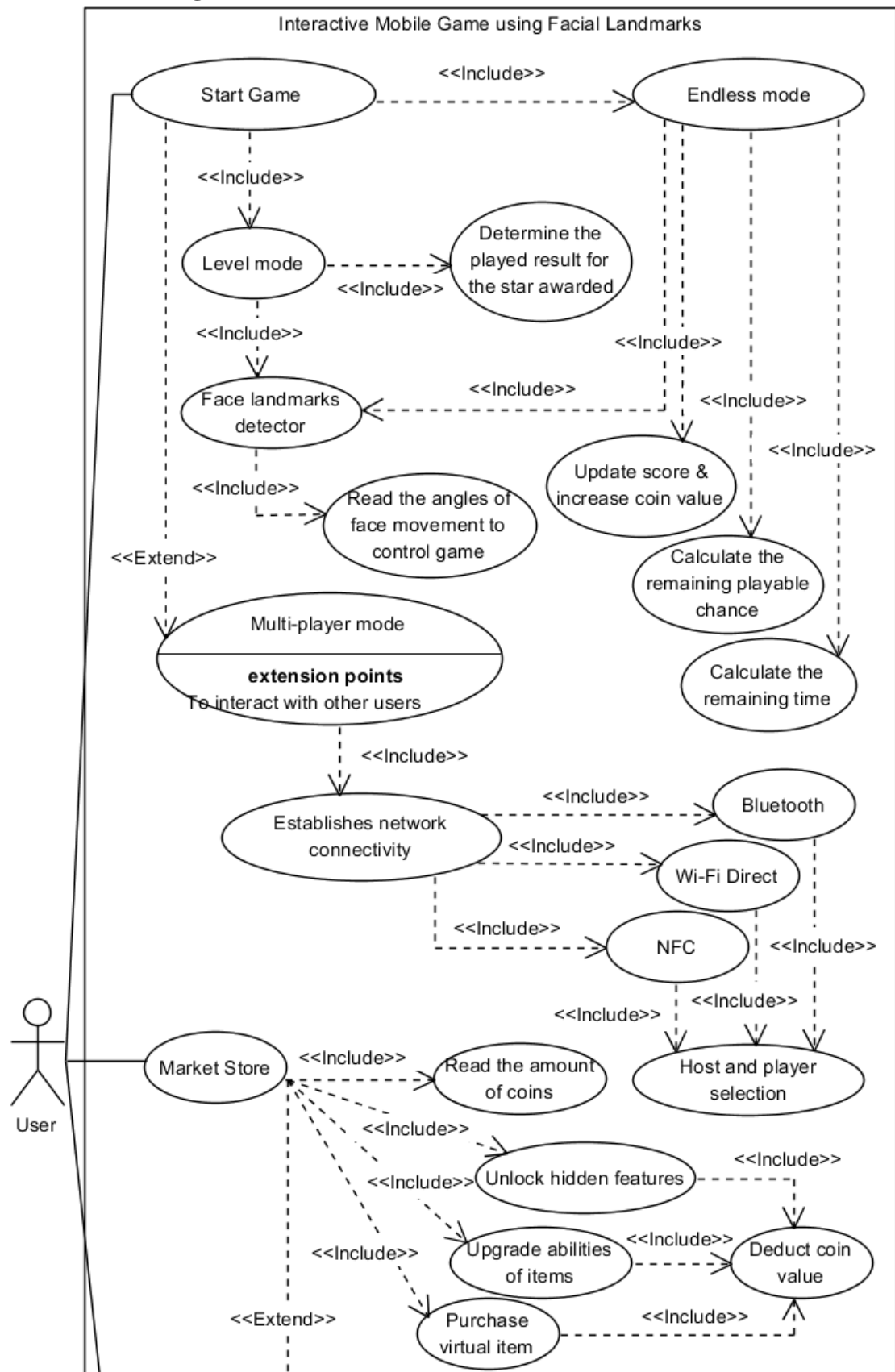


Figure 3-3: Upper half of use case diagram

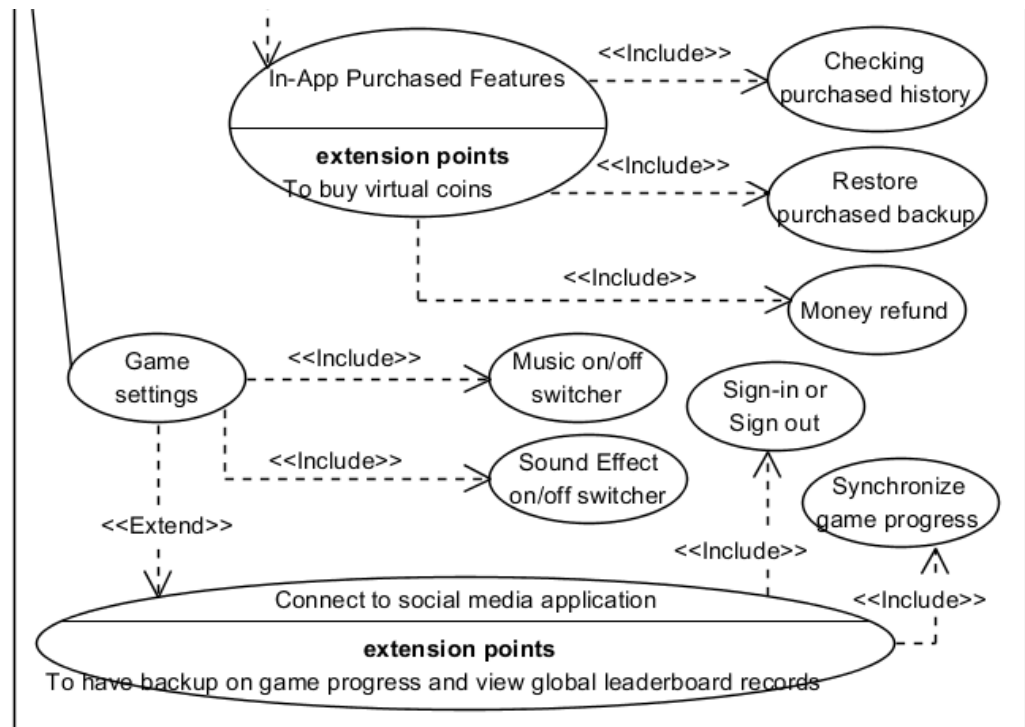


Figure 3-4: Remaining half of use case diagram

**Use case diagram description:** There are single player mode and multi-player mode that available to user. In single player mode, user can enjoy both endless mode and level mode. Endless mode will require to calculate remaining playable chance and remaining time, to update high score and to increase amount of coins. Besides, level mode will require user to determine the played result in term of high score in order to figure the achievement awarded for each game level. On the other hand, in order to activate multi-player mode, user is required to enable some kinds of network connectivity like Bluetooth, Wi-Fi direct and NFC based on the specification of devices. User is required to select for being host or normal player before proceeded further. In order to play the game, user has to ensure his or her face is able to be captured within the front camera. If there is any face movement or any special face expression from user, then user may able to trigger some controls in the game. Moreover, user can purchase virtual item, upgrade ability of item and unlock hidden game features using the coin amount available. User may have in-app purchased option in order to obtain coins quickly. User can check purchased history, restore purchased backup and refund money from in-app purchased option. Furthermore, user can adjust the volumes of music and sound effect from game settings. Lastly, user can connect to social application for synchronizing the game progress and viewing high score from both local device and associated server.

### 3.4 Activity Diagram

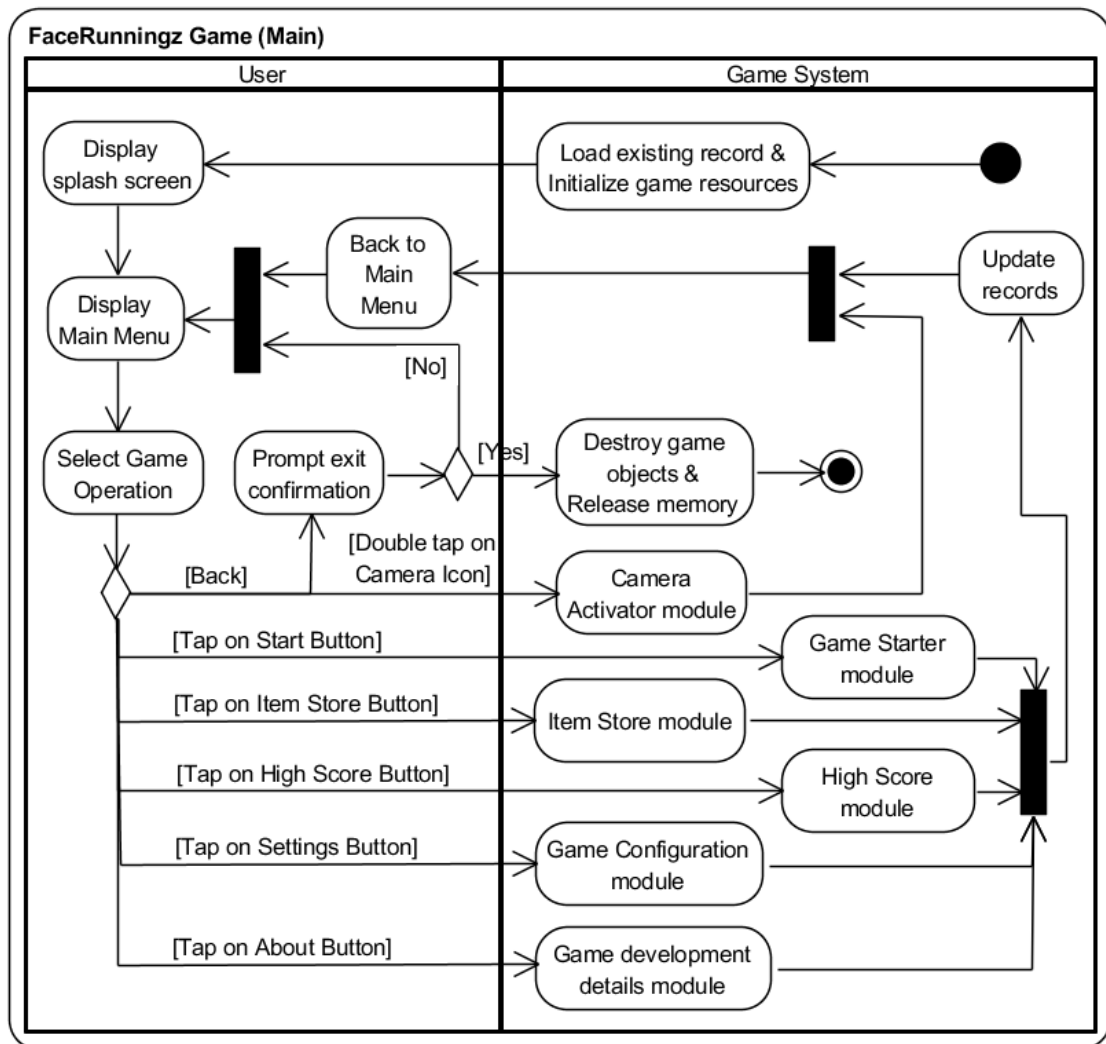


Figure 3-5: Game System (Main) Activity Diagram

This activity diagram is the overall diagrams that consists of all the modules in the game system. Once the game has been executed, it will display a splash screen for several seconds before proceeding to main menu. Then, user can select different modules in order to start the game, customize the game objects, select various game modes, check high scores, change game settings, and unlock testing mode. Every single module will require to update their own records after executed in order to prevent data loss issues. Firstly, game starter module is used to construct all the game objects in game scene based on game settings and user selection, control and manipulate the game objects based on user input and calculate high score and credits obtained. Secondly, camera activator module is used to detect facial landmarks from users' face, read facial expression and calculate the movement values for x-axis and y-axis before send the

message back to the game system. Thirdly, item store module is used to handle all the item purchased transaction and game object customizations based on items availability, items' costs and current amount of credits that users obtained. Fourthly, high score board module is used to display various high scores based on different playing method and game scene. Fifthly, game configuration module is used to make some changes on game settings based on user own preferences. Lastly, game development detail module is just used to display developer details but it contains a hidden testing mode which includes advance game settings. It mainly used for testing purpose otherwise users will be lost fun from playing the game.

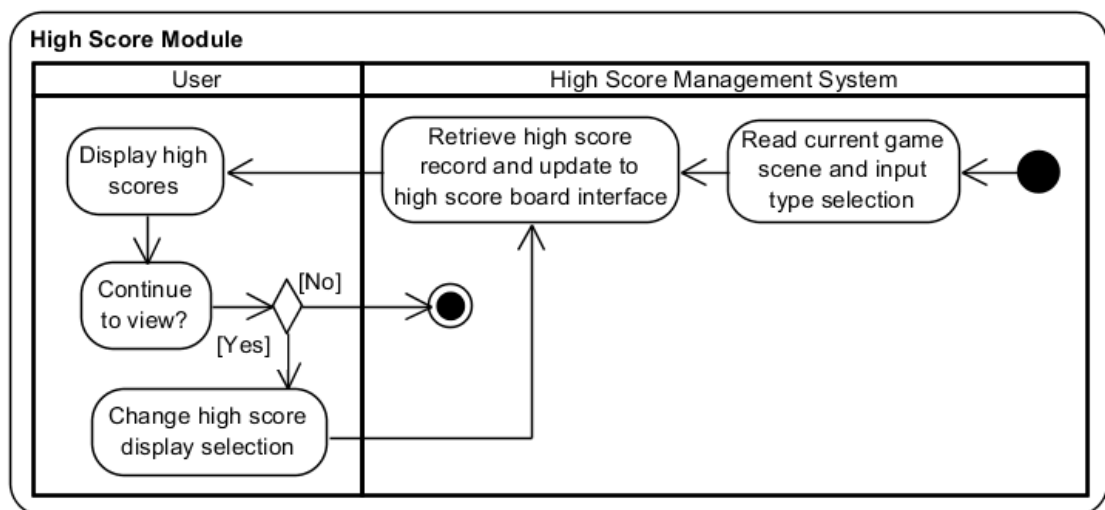


Figure 3-6: High Score Module Activity Diagram

This module is used to manage, view and display high scores that user obtained and achieved. Once high score board module has been executed, the system will retrieve relevant high score data and update them to the high score interface. It will continue to retrieve and update if user makes any change on the high scores.



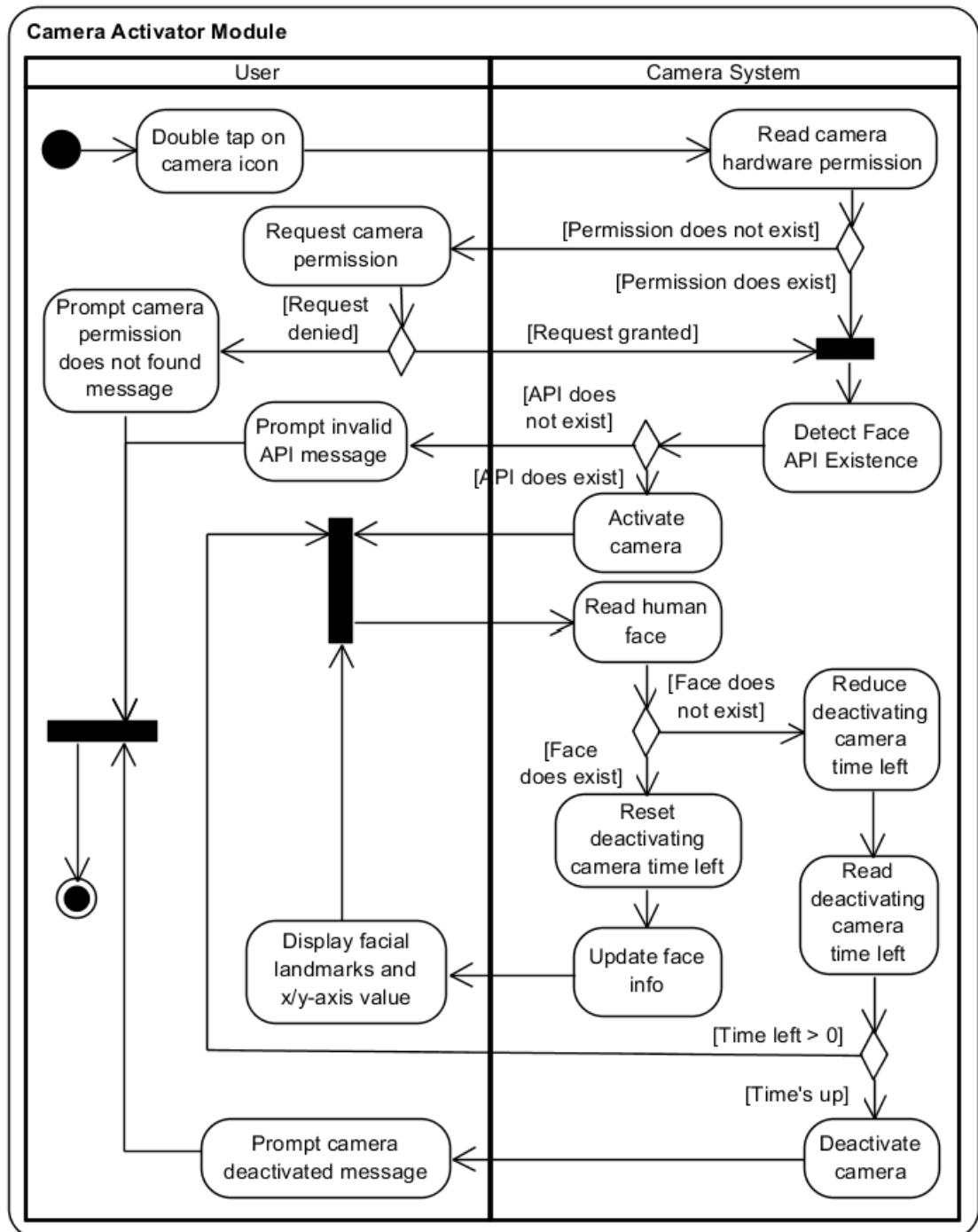


Figure 3-7: Camera Activator Module Activity Diagram

This module is used to manage and detect facial landmarks from human face. Basically, it will only read and retrieve facial information from one of the nearest faces. Once the module has been accessed, it will request camera hardware permission from user for the first time. If user has granted this permission before, then this step will be skipped. After that, it will detect existence of Google Framework inside the device. If the API cannot be found, then an error message will be prompted up, otherwise this step will be

ignored. Once the camera has been activated, then it will try to seek for human face. If there is any face that has been detected, then it will read facial information such as facial landmarks, face position in term of x-axis and y-axis and face expression and store them into face storage while waiting for the game system to retrieve. In case, if human face cannot be detected, then the face storage will be reset to default. Basically, the face detection speed is depended on frame per second (FPS) which has set up inside the backend of the program. In order to save resources of CPU, GPU and RAM for minimizing the usage of battery, the system will restrict the active period of camera. For instance, the camera will switch off automatically if the system cannot detect human face for 10 seconds. Also, user can switch off the camera manually via double tap on the camera display.

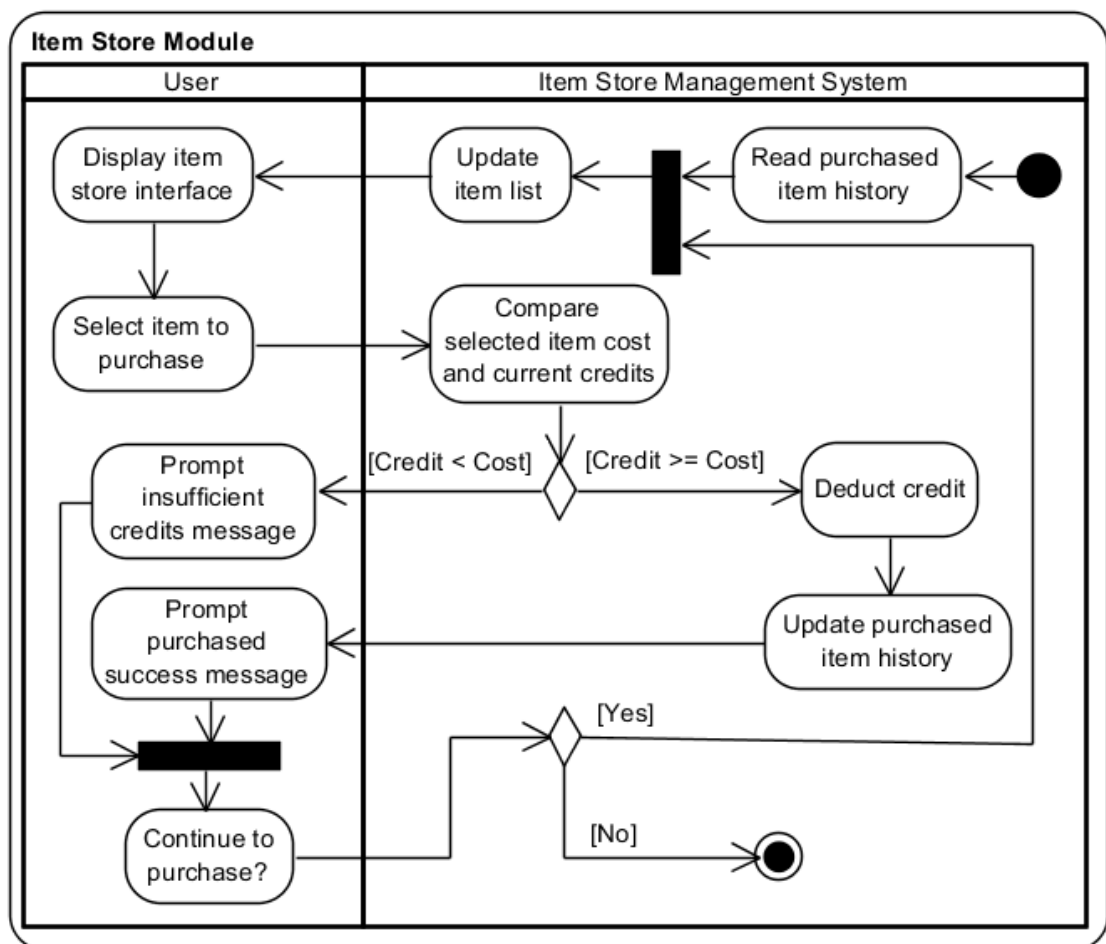


Figure 3-8: Item Store Module Activity Diagram

This module is used to manage, purchase, select and switch game items from item store. Basically, the item store consists of different types of game objects which allows user

to switch and extra modes in term of credits multiplier, shooting mode and shield mode. Once the module has been executed, the system will read the item purchased history and update item store interface. If user wants to make purchase, then the system will go through the cost of selected item and the credits amount that user obtained. If user has enough credits then he or she can buy the items, otherwise the system will prompt up an invalid item purchased message to user. Once the transaction has been approved, then the item will store in the purchased history and the credits will be deducted. These records will be written in external storage for the system to retrieve next time.

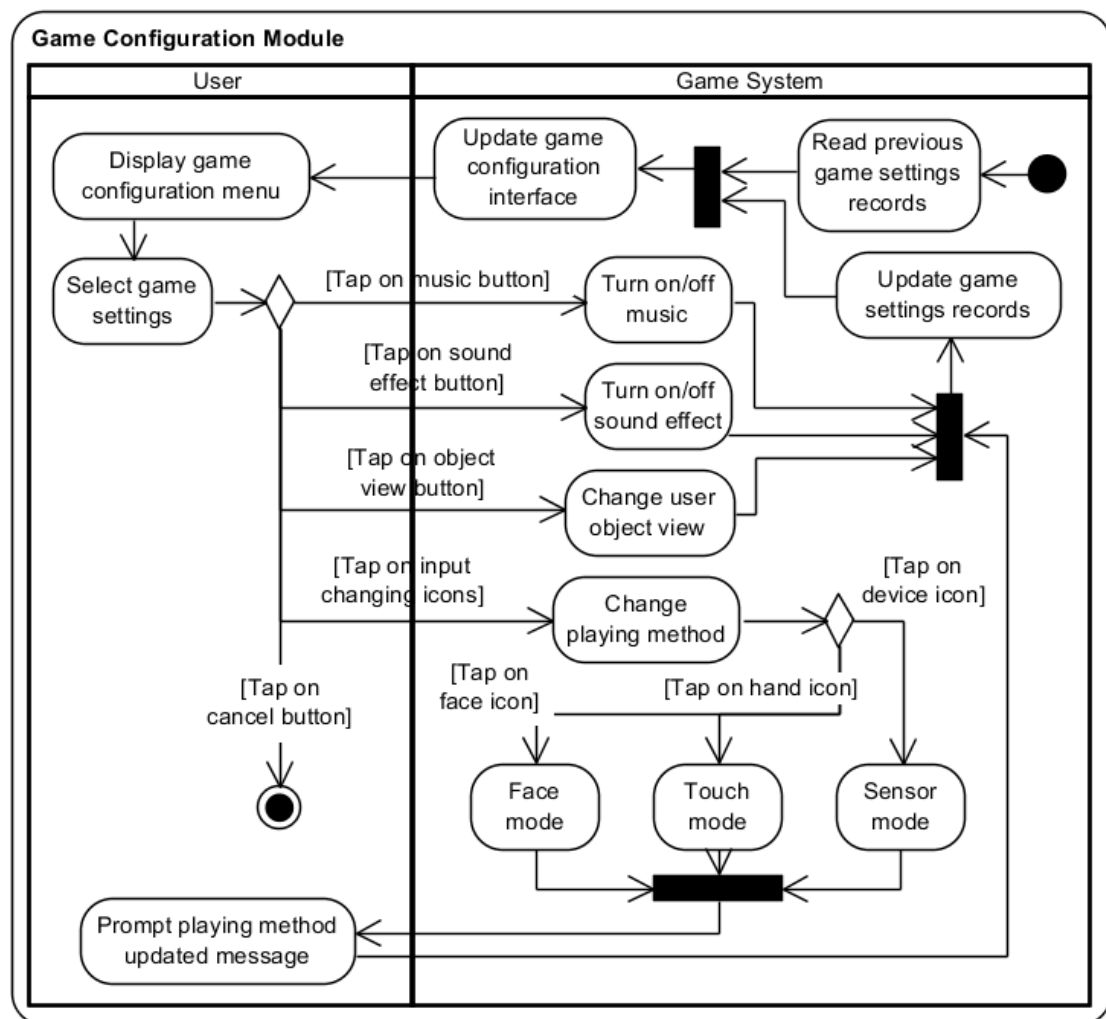


Figure 3-9: Game Configuration Module

This module is used to manage and update game settings. User can adjust volume of music and sound effect, change object view in term of first person view and third person view, and change playing method. Basically, the system will update immediately once the user changes settings.

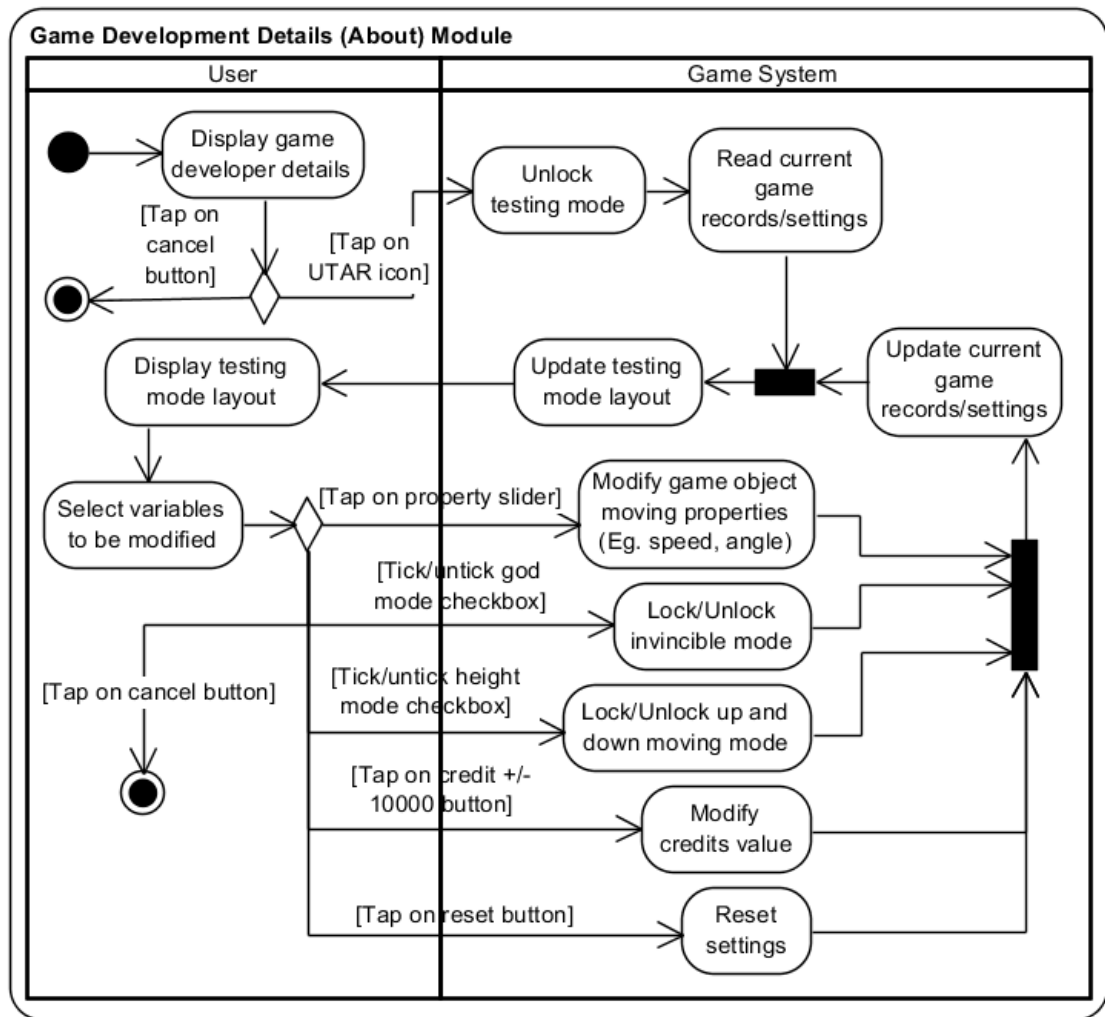
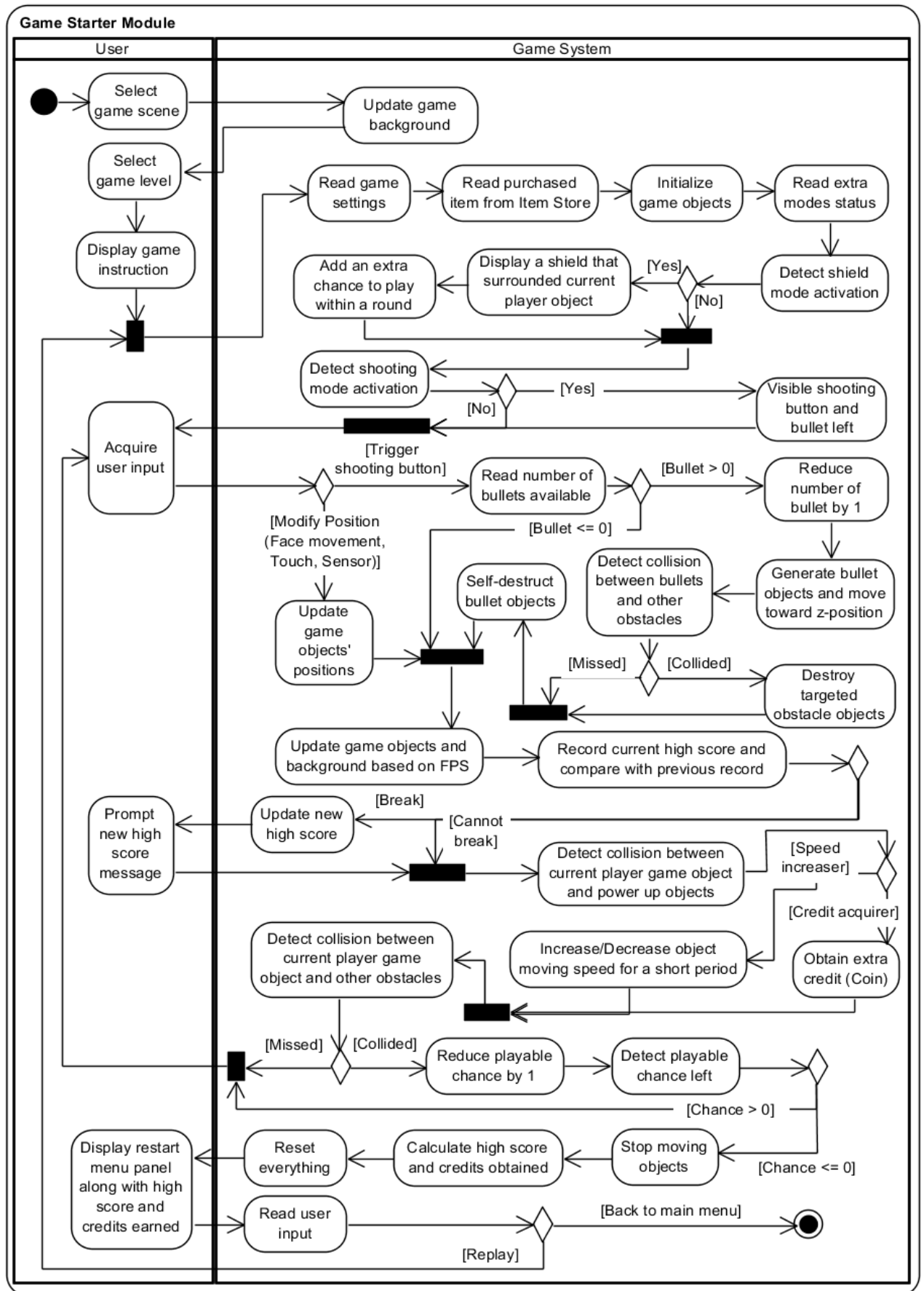


Figure 3-10: Game Development Details Module

Initially, this module is used to display the game development details like developer to user, but it included testing mode that required to unlock. Basically, the testing mode allows developer to make some additional changes such as modifying credits, enabling invincible mode that allows to move game object by ignoring obstacles, modifying game object moving properties and unlock other extra modes. Regarding to the module flow, it is similar to game configuration module. At the first time, it will load the current game settings before it displays the interface to user or developer. It will update game settings if there is any new change.



*Figure 3-11: Game Starter Module*

This module is used to construct all the game objects in game scene based on game settings and user selection, control and manipulate the game objects based on user input and calculate high score and credits obtained. Once user has selected the game scene and the game level, then the game system will initialize the game scene and construct the game objects based on game settings. If there is any extra mode available, then the system will activate and consume it directly. After that, the system will move the game object based on user input. If shooting mode is activated, then it allows the game object to fire the bullets in order to destroy those obstacles. If shield mode is activated, then user can have another chance of playing the game within a round after the game object crashed with obstacle. Also, user can pause the game within the module. Once user is running out of playable chance, then the system will display the calculated high score and credits obtained. At this moment, user may choose either to restart or to exit the game.

## 3.5 UML Diagram

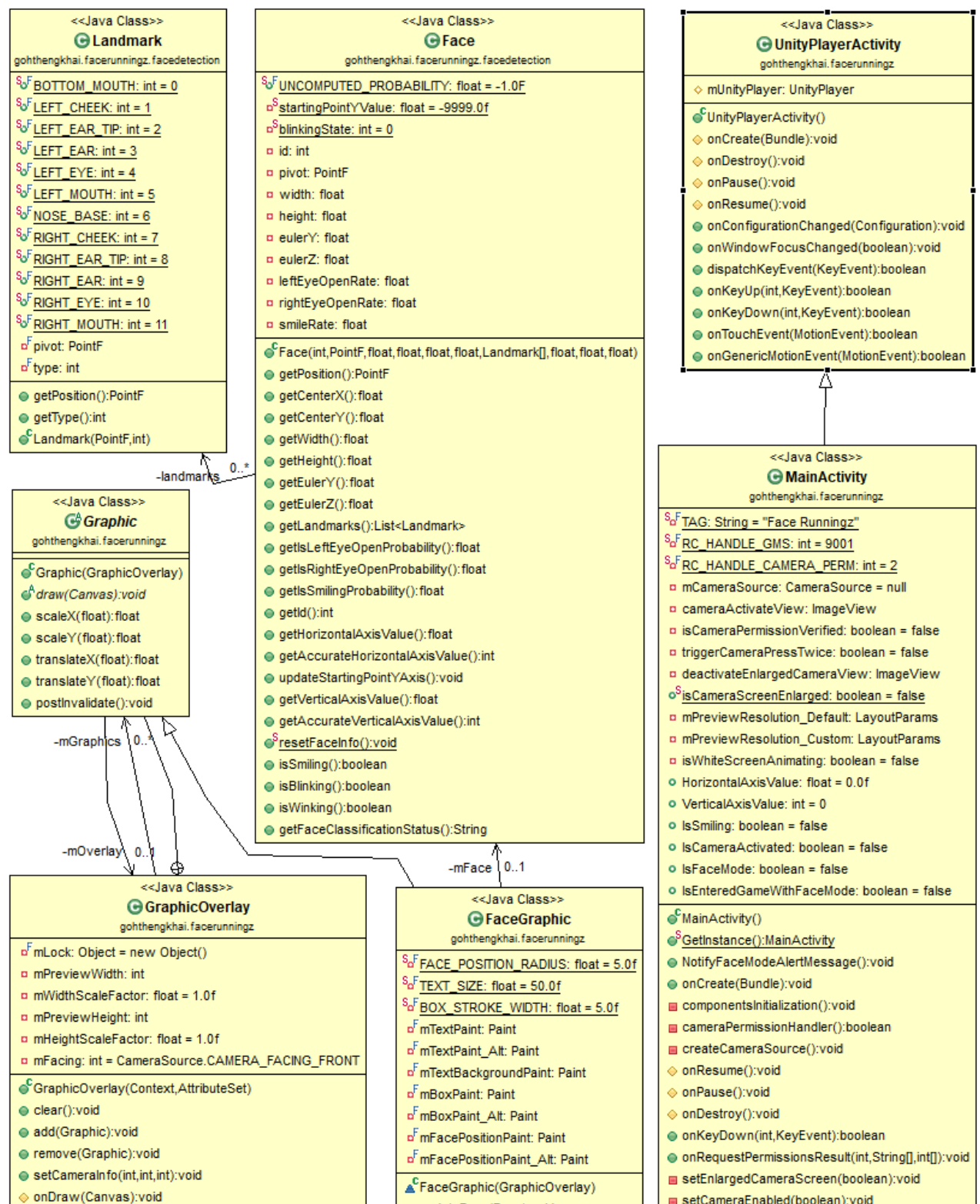


Figure 3-12: Upper half of UML diagram for Face API's backend (Java)



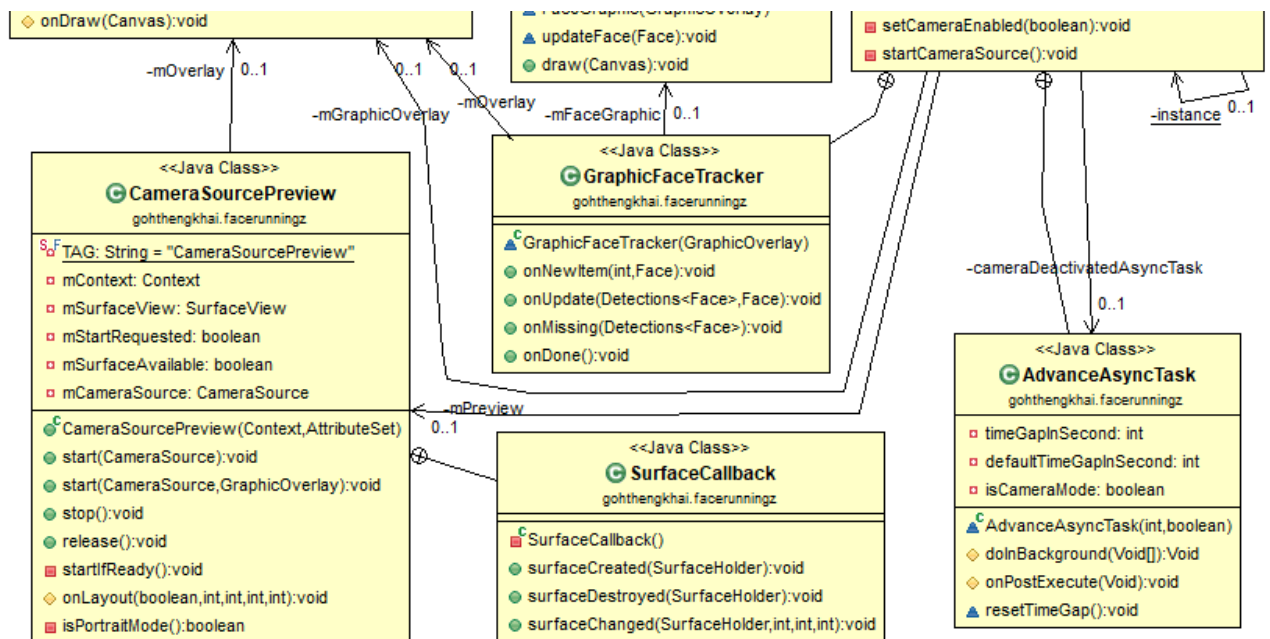


Figure 3-13: Remaining half of UML diagram for Face API's backend (Java)

Figure 3-12 and figure 3-13 show that the relationships among all the classes for face API's backend in Java language and their relationships included inheritance, polymorphism, aggregation and abstraction. Basically, Face class and Landmark class will be used in storing and managing facial information. Landmark class is just a part of Face class since it can only store a series of facial landmark information within face object. They will be used to pass message and communicate with Google Mobile Vision API via Graphic Face Tracker class. Once the face detection connection has been established, the API will send the face object back to the system. Usually, the retrieved face object will be an immutable object so the API can ensure the object consistency by disallowing third party to modify once the object has been constructed. Besides, Graphic class, Graphic Overlay class and Face Graphic class will be used to draw customizable shapes and texts on the camera display which can be known as low level canvas. For instance, developer may simply change a few lines of codes to perform AR technique on human face. Moreover, Camera Source Preview class and Surface Callback class will be used to manage the camera display with the relevant screen resolution. In addition, Main Activity class is a subset of Unity Player Activity class where they will be to manage the overall flows of the face detection program. Lastly, Advance AsyncTask class can allow a small program or a mini module to run on background so it can save time and resources to perform other actions within the same period of time.



### 3.6 Class Diagram

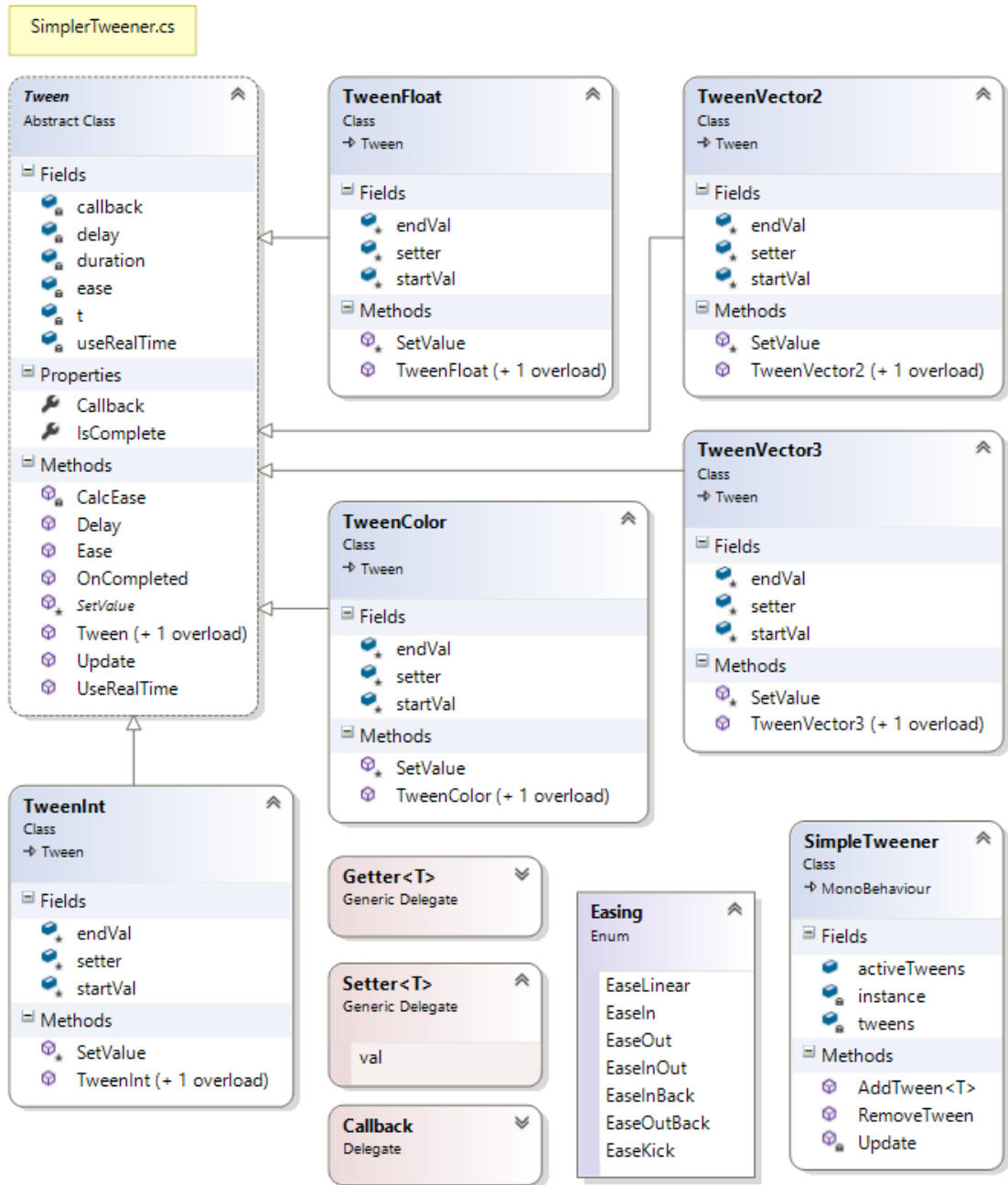


Figure 3-14: Simpler Tweener Class Diagram

Basically, all the above classes are written within a C# class file. The purpose of these classes is to create and make a simple 2D or 3D animation within a system. For instance, if developer wants to tween an object's position over 5 seconds and to delay by 2 seconds before calling a custom method, he or she can refer to the code below.

```
SimpleTweener.AddTween(()=>transform.position, x=>transform.position=x,
    targetPosition, 5.0f).Delay(2.0f).OnCompleted(()=>CustomMethod());
```

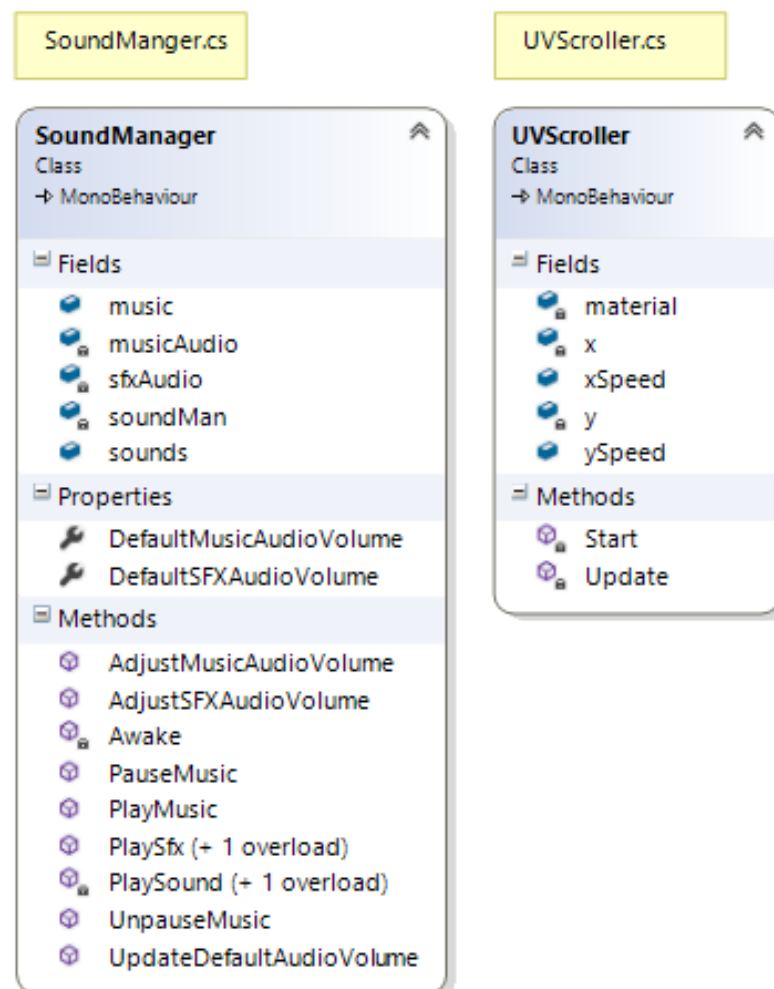


Figure 3-15: Sound Manager and UV Scroller Class Diagram

Sound Manager class is used to unite game background music and sound effects together so that it will simple and convenient to manage within a class without writing duplicated code. Basically, this class enables developer to manage a couple of audio path lists before the system able to play audio. Also, the volume of background music and sound effect can be adjusted independently. For instance, in order to setup and play a music, apply *SoundManager.PlayMusic("Audio\_File\_Path");* directly. Thus, it can reduce the project complexity and workload of developer.

Furthermore, UV Scroller is used to scroll UVs of a model in specific speed and direction. Once the script has been executed, it will automatically link with the game object path and update the texture based on UV animation rate.

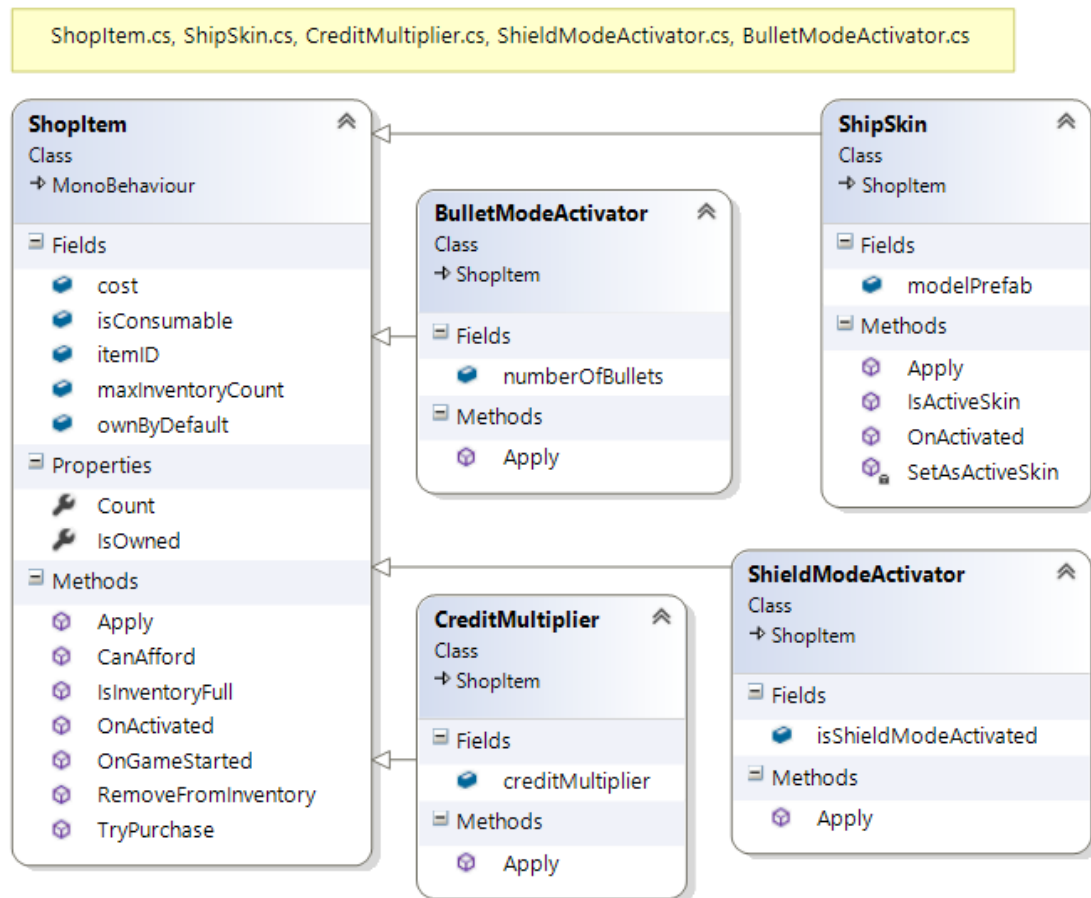


Figure 3-16: Shop Item Class Diagram

Shop Item class is used to manage a list of items in item store where each item contains its own properties such as id, cost, number of duplication and others so developer will be easy to use by creating a new class and inherit to Shop Item class. At this moment, there are 4 subclasses which can perform different outputs since they shared certain game states only. Firstly, Ship Skin class is used to change the appearance of game object. Secondly, Bullet Mode Activator class is used to unlock shooting mode where it allows user to destroy the obstacles. Thirdly, Shield Mode Activator class is used to enable an extra protection so user can have another chance of playing the game within a round after he or she crashed the obstacles for the first time. Lastly, Credit Multiplier class is used to increase the credits obtained by multiply the value with 2 or 3 times in a round of the game.

Powerup.cs, SpeedBoost.cs, SlowMotion.cs, BonusCredits.cs, PowerupManager.cs,  
PowerupCollectable.cs, PowerupSpawner.cs, PowerupSpawnPoint.cs

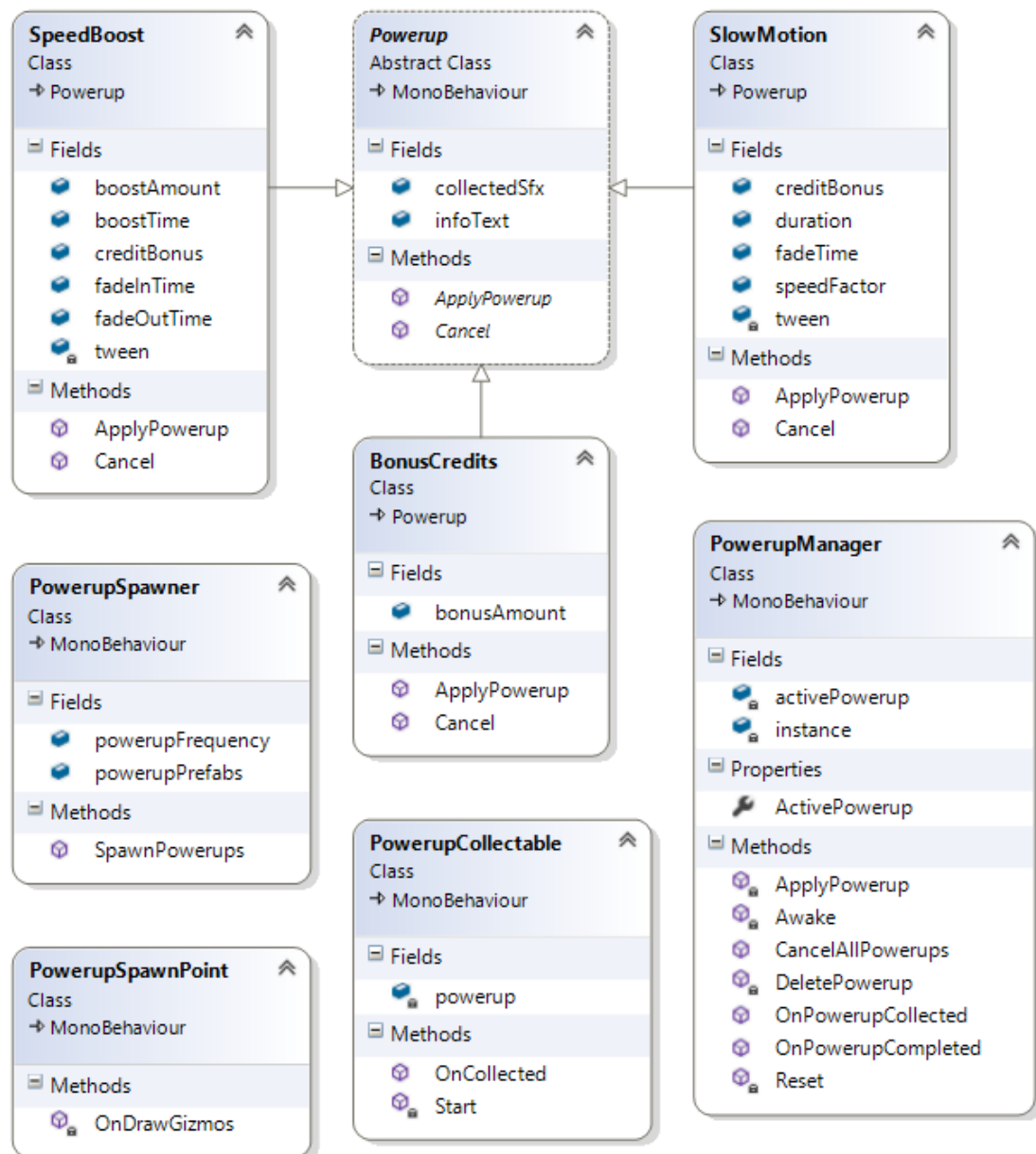


Figure 3-17: Power-Ups Class Diagram

These power-up classes are used to construct power up objects and set them at a random position within game scene. Once the player's game object intersects with the power-up object, it will active certain special abilities like slowing or speeding up the player object's movement and gain some extra credits within a few seconds. Thus, it can enhance the game challenge to user.

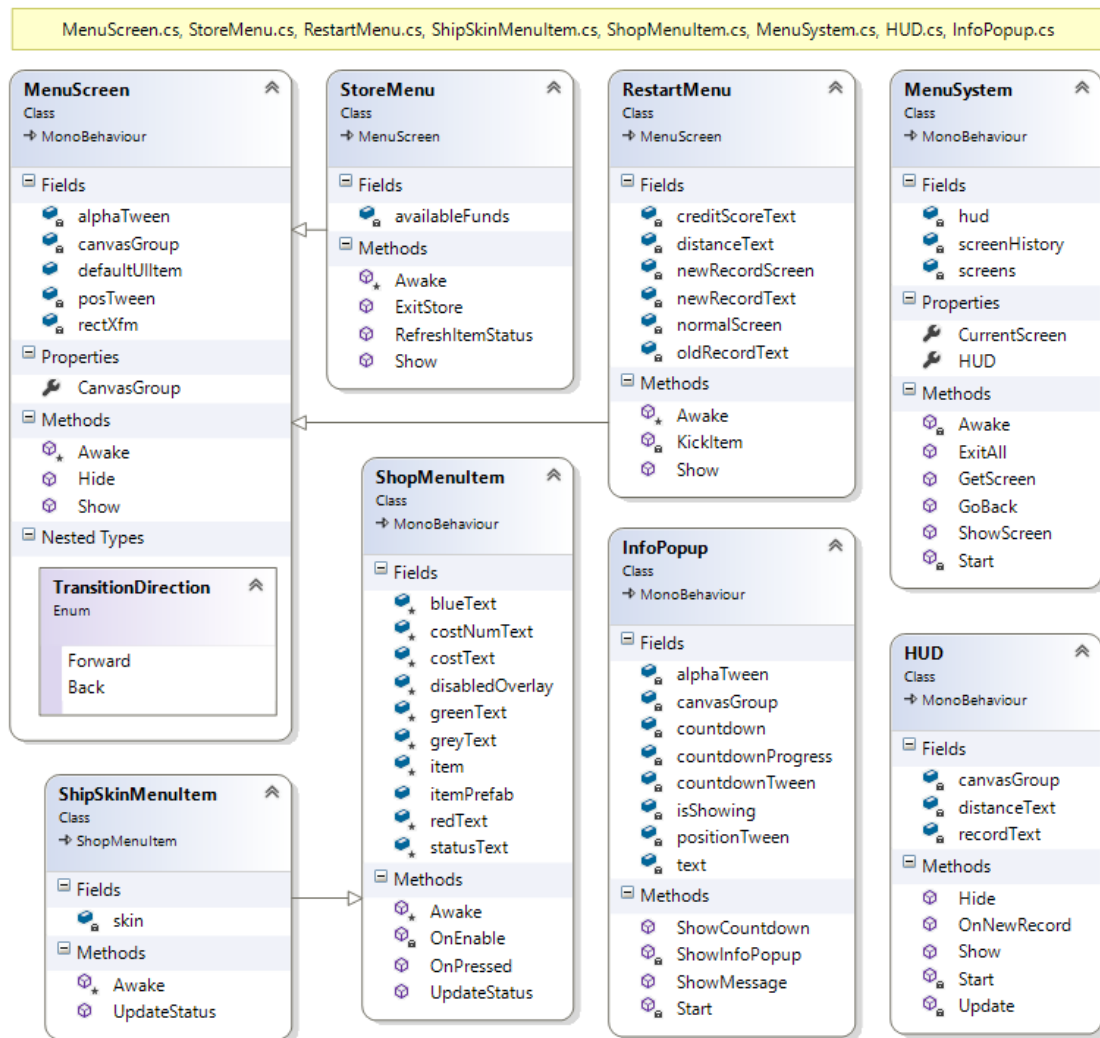


Figure 3-18: Menus Class Diagram

The Menu System class is the main class to handle other menu classes. Basically, there is a stack list to manage these menu classes. If the user chooses to enter another menu, then a new menu reference will be stored and push inside the stack. Once user left the menu, then the system able to remove and pop back the previous screen menu. The other menus are constructed based on the need of their interfaces respectively.

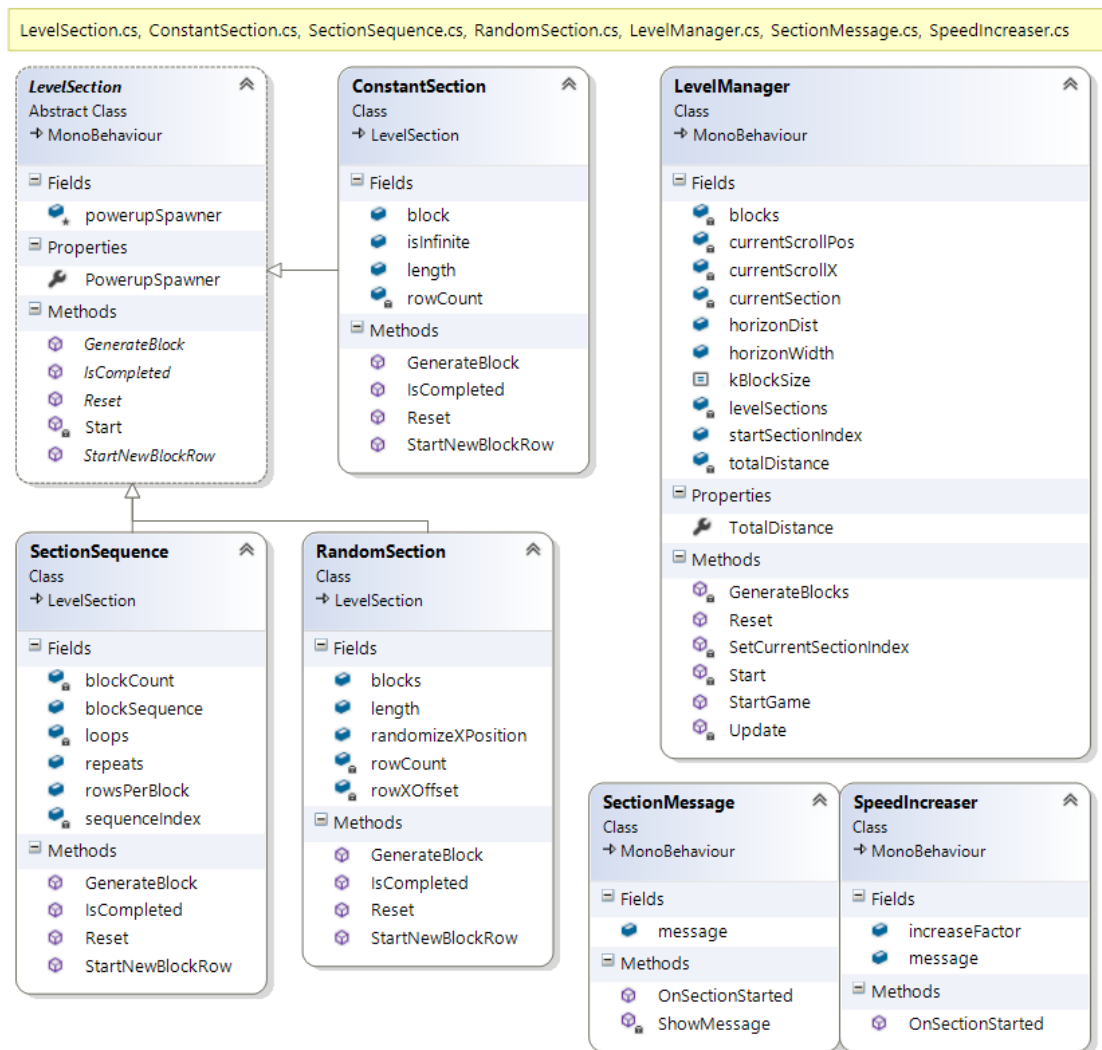


Figure 3-19: Level Generations Class Diagram

Level Manager class is responsible to generate all the blocks and move them through the player's game object. Once the blocks are out of the player view, then they will be destroyed and remove from the game scene. Normally, each block section has a specified length so once the section has reached the limit length then it will proceed to another section. If it has reached the last section then it will loop back to the first scene and keep on repeating until the game ends. Constant Section class handles blocks at constant position, Random Section class handles movable blocks and Section Sequence class able to arrange the block in a nice sequence. Also, Section Message class is used to alert user once it reached certain blocks while Speed Increaser class is used to alert the speed up notice while increase the movement speed of player's game object.

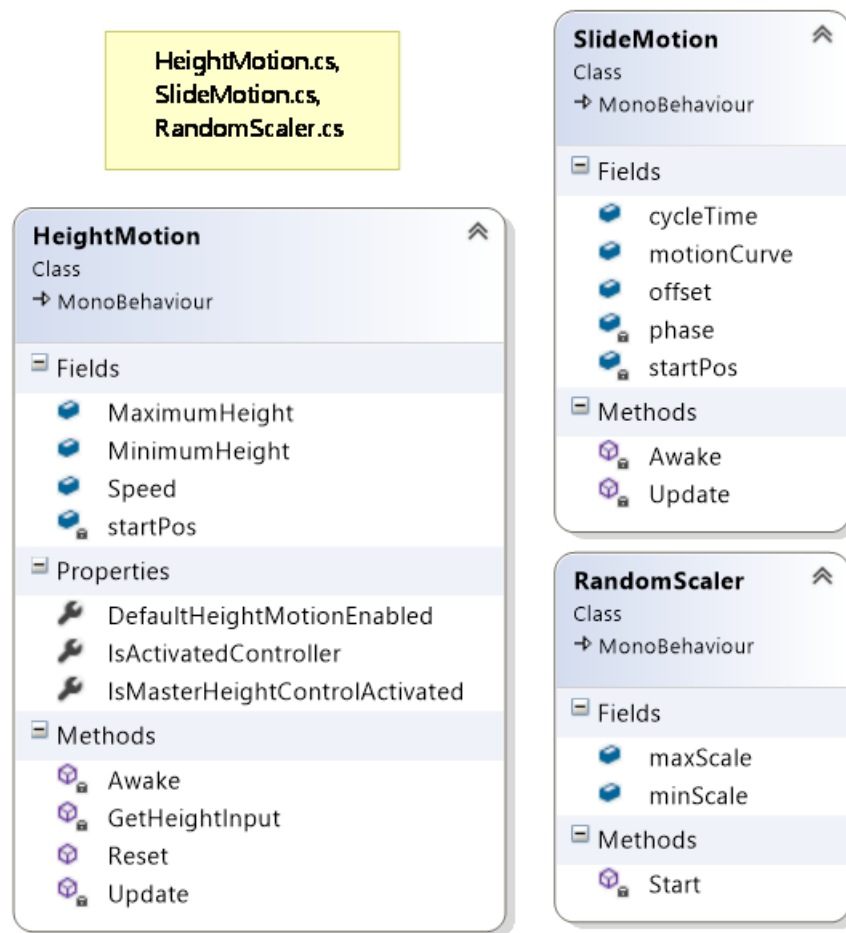


Figure 3-20: Height Motion, Slide Motion and Random Scaler Class Diagram

These above classes are able to interact with the player's game object in game scene. Firstly, Slide Motion class handles the player object movement in terms of x-axis while Height Motion class handles y-axis object movement. Besides, Random Scaler class is used to generate objects with random size in terms of x-axis, y-axis and z-axis but the size will be restricted to specific length, width and height for avoiding from generating too large or too small objects.



GameManager.cs, PlayerControl.cs, SceneSelector.cs

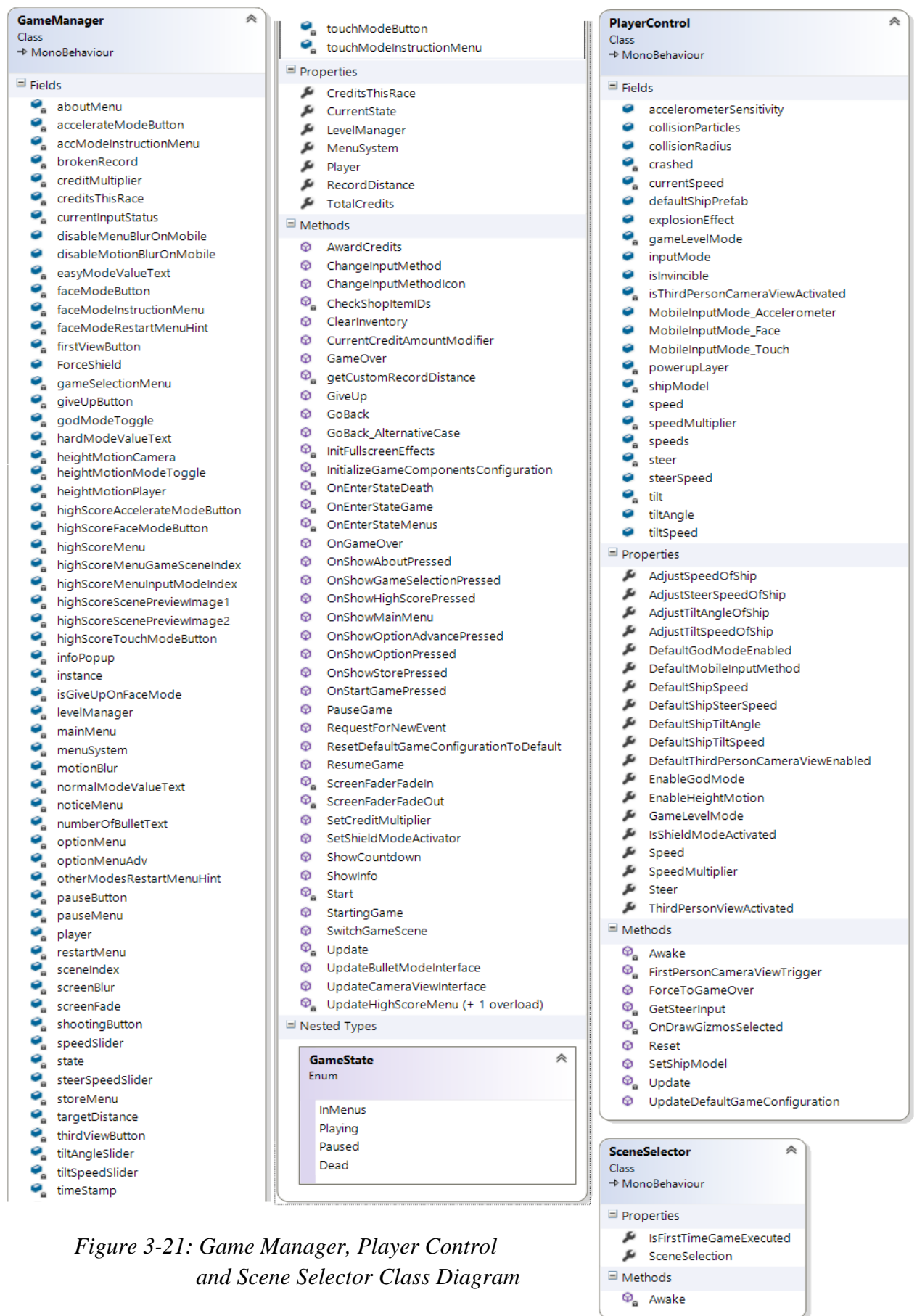


Figure 3-21: Game Manager, Player Control and Scene Selector Class Diagram



According to figure 3-21, it shows that Game Manager class, Player Control class and Scene Selector class. Game Manager class is the game core where it manages and links almost every classes and game behaviours together. Moreover, Player Control class is used to retrieve user input in term of face movement, touch position and sensor movement and calculate the movement speed based on a complex algorithm before these values pass to Slide Motion class and Height Motion class (figure 3-20) to perform further operation. Furthermore, Scene Selector class is used to determine the sequence of game scene that able to load at first so that the system will not have any conflict on switching among game scenes.

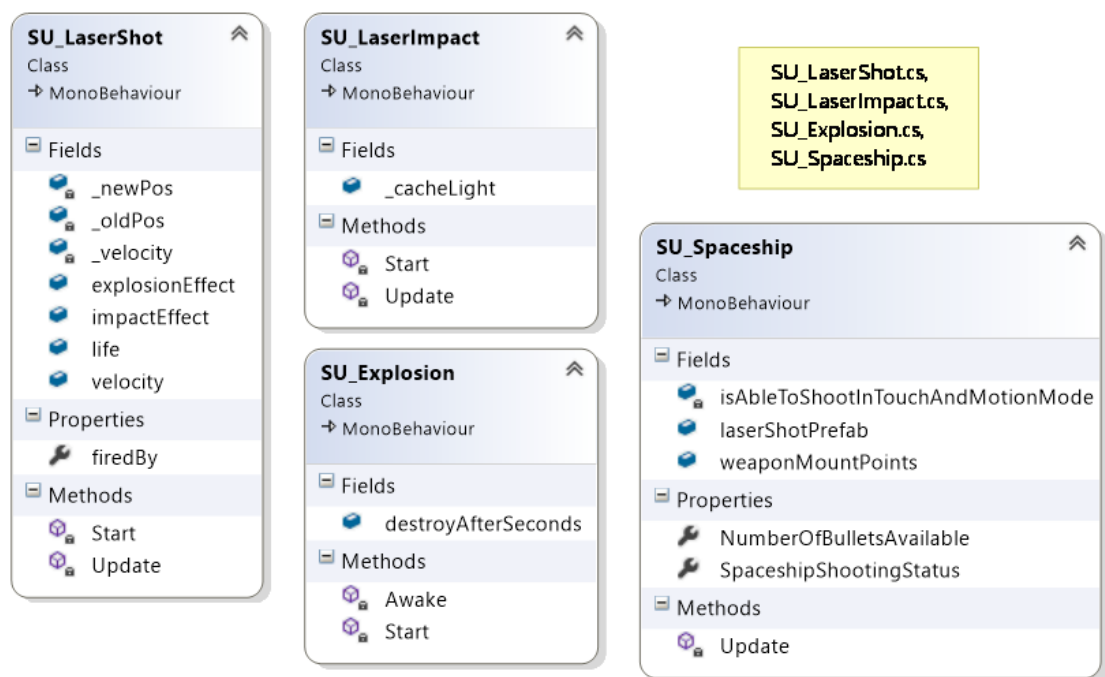
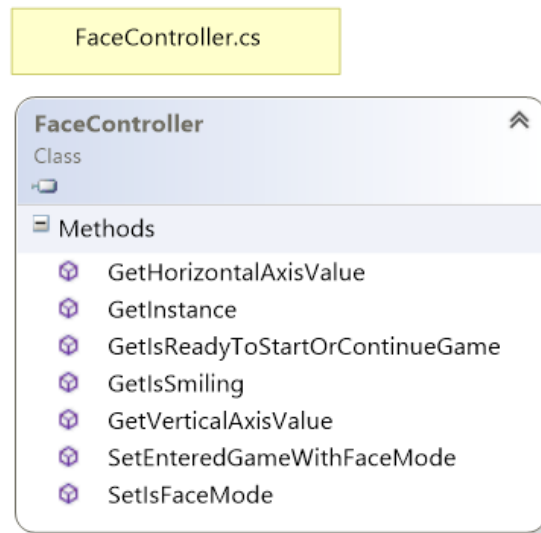


Figure 3-22: Spaceship Battle Class Diagram

The above classes are used to generate some bullet objects, calculate the bullets' speeds, determine the collision between bullets and other object obstacles and create an explosion effect if the collision happened.



*Figure 3-23: Face Controller Class Diagram*

Face Controller class is purposely created to communicate with Java source code since Google Mobile Vision Face API only available in Android and iOS while Unity game engine requires C# or JavaScript scripting language as its backend. Basically, it will retrieve all the facial information from Java backend and pass them to Player Control class (figure 3-21) before Slide Motion class and Height Motion class (figure 3-20) execute.

### 3.7 ERD Diagram

Figure 3-24 shows that the expected entity relationship diagram that used to store the game data. There are 6 tables, including account, user, game progress, high score and configuration that are linked together through an important key that names User ID.

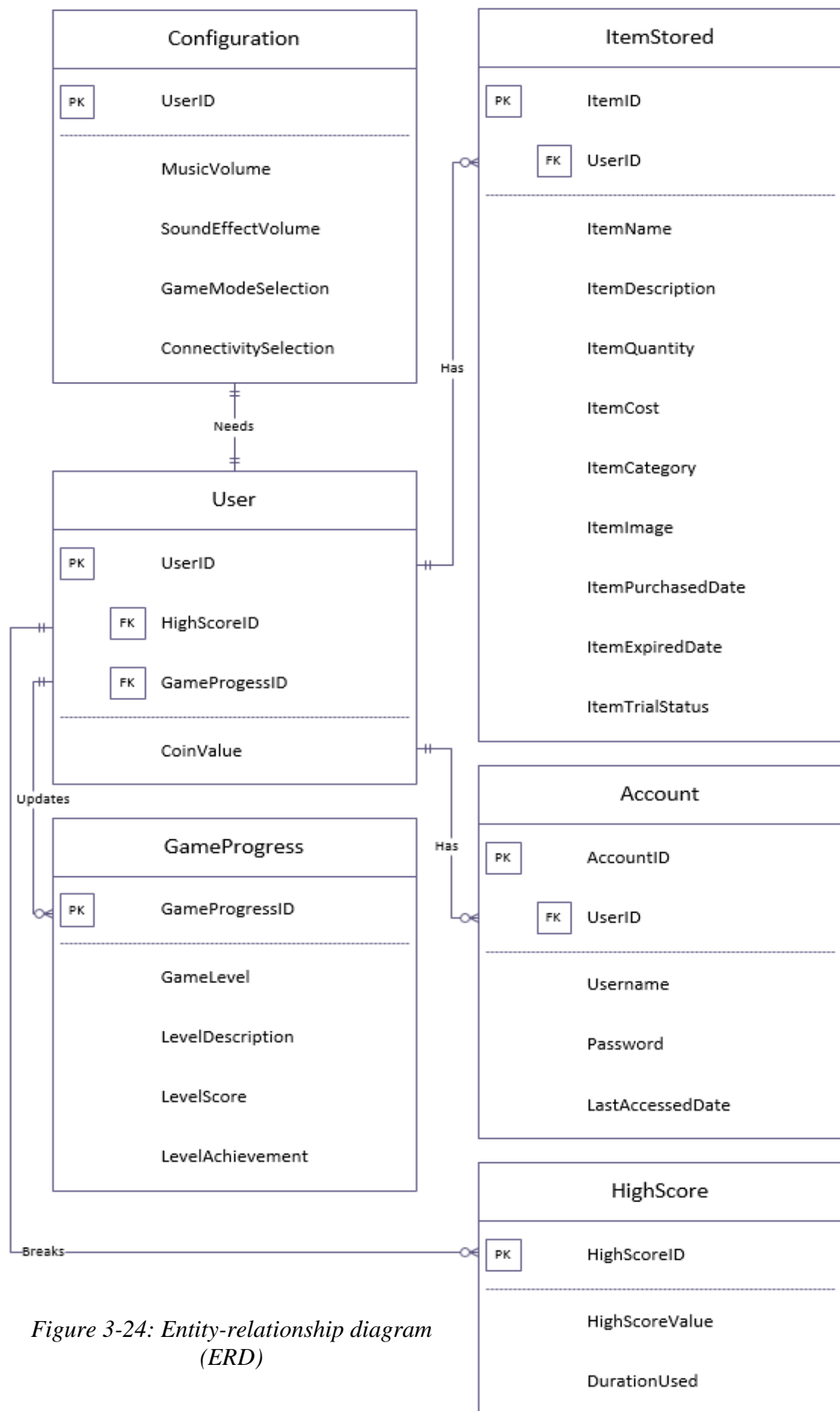


Figure 3-24: Entity-relationship diagram (ERD)

### 3.8 Low-Fidelity Game Prototypes

These low-fidelity prototypes are developed to gain some idea and concept for better initial project visualization. They usually will be abandoned or re-modified after a while so they have least chances of being final program prototypes.



*Figure 3-25: Main Interface*

*Figure 3-26: Item Purchased Interface*

According to figure 3-25, it is obviously a game main menu and briefly describes about all the interconnections of game layouts respectively. Besides, figure 3-26 provides visualization on market store where user can unlock some virtual items and upgrade their abilities by purchasing them based on current amount of coins. The game will provide in-app purchased option for those who do not have enough coins but wish to obtain them quickly.

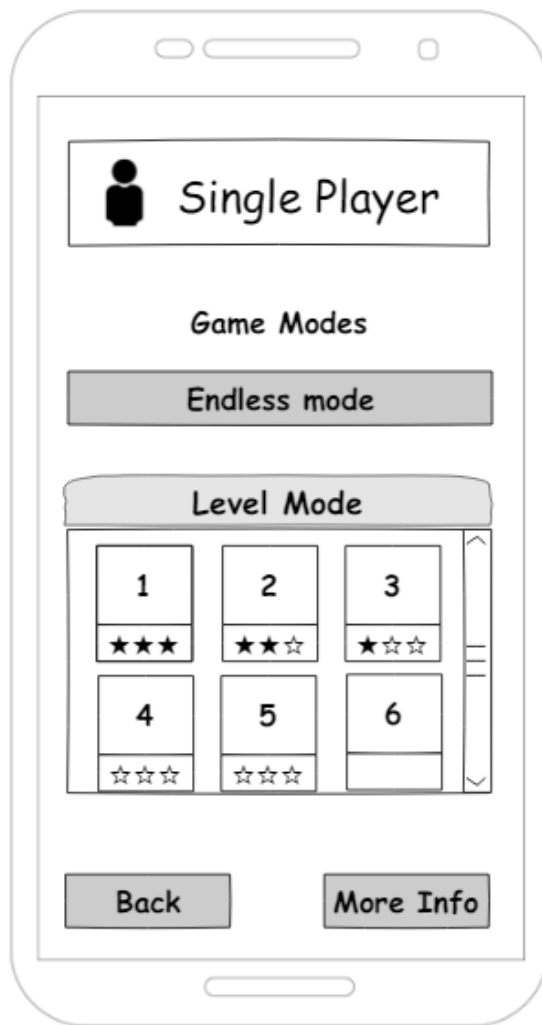
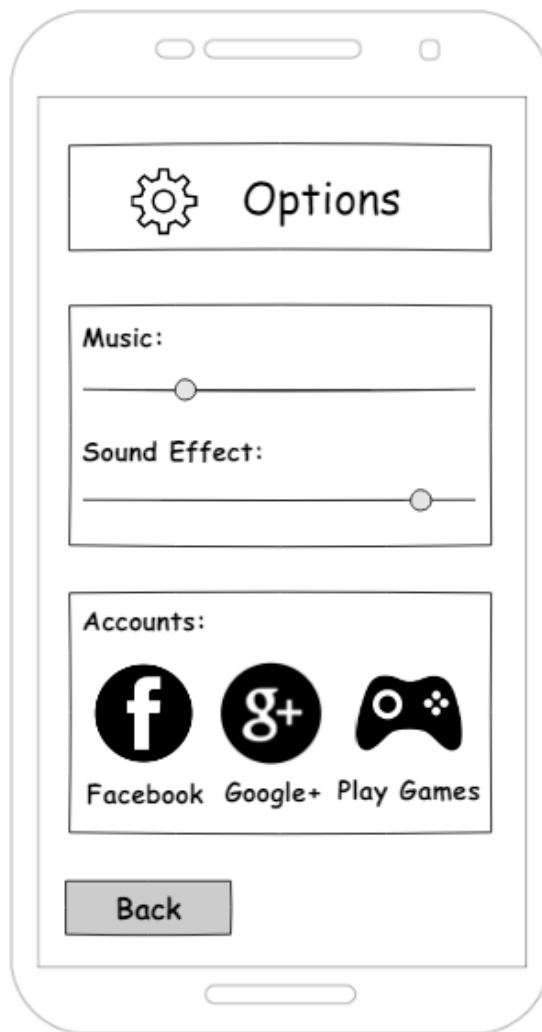


Figure 3-27: Single Player Interface

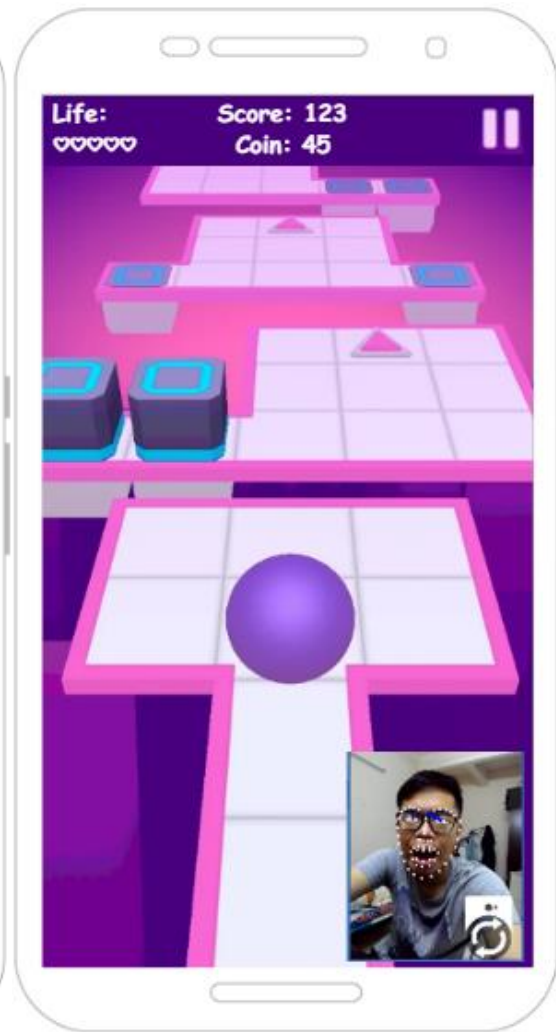


Figure 3-28: Multi-Player Interface

Single player mode (figure 3-27) as well as multi-player mode (figure 3-28) are available in the game. By default, the game is based on the remaining time and remaining playable chances to determine the ending condition per game in endless game mode. For level game mode, rewards will be given to users if they reached a certain high score. User may obtain some coins through the game. Meanwhile, multi-player mode performs exactly the same game modes as endless mode but with some additional features using Bluetooth technology where users can interact with each other. User can choose to be either host or guest. Once the number of players is enough, the game will be started.



*Figure 3-29: Settings Interface*



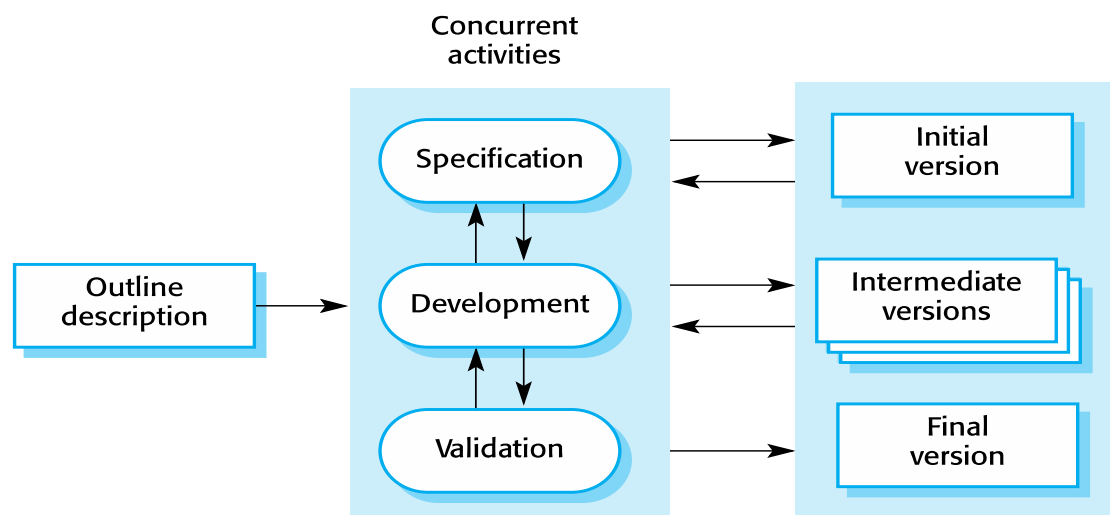
*Figure 3-30: Gameplay Interface*

User make some configurations on game regarding to the volumes of music and sound effect. Also, they can synchronize their game progress and view high scores of their friends by login to some associated social media accounts. Furthermore, the front camera will be activated to read the facial landmarks for determining the movement of the game object (ball) as it is shown in figure 3-30.

## Chapter 4: Design Specifications and Implementations

### 4.1 Methodology

The agile development where it is classified under incremental approach is selected and will be applied in the proposed game since it is not a large or complex application. Basically, this approach does not require too much of detail documentations so the cost of accommodating change reduced. Besides, it allows the prototype to be developed and tested incrementally as separate components. The components then integrated as one complete application which can fulfilled all requirements. It actually breaks development into many iterations which eases the development process and constantly comes out with small release that support for rapid user feedback and continuous application refinement. Refactoring is encouraged to enhance the quality of code and make it more understandable and readable so that it can speed up the development process.



*Figure 4-1: Incremental Development Model*

## 4.2 Software and Hardware Requirements

### 4.2.1 Software Requirements

#### 1. Android Studio

Google Inc. has no longer supported Eclipse since year 2016, Android Studio is the only official IDE that used for Android application Development. Currently it can use to build any version of Android application where it supports up to Android 7.1 (Nougat). It is also known as a world-class code editor as all the syntax errors and program errors can be detected and debugged quickly. Besides, there are a lot of frameworks and libraries that can be obtained easily from internet since majority of them are considered free. Moreover, developer can create most of the static layouts by using drag and drop features where its coding part can be directly linked with interface id in order to trigger some kind of program activities.

#### 2. Unity

Unity is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites. It can use to create any 2D or 3D game environment by attaching those beautiful and high-optimized graphic resources from its asset store where they can be either free or required to pay based on users' preferences. Since all the graphic resources are scripted with both C# and JavaScript languages, user can transform the behaviours of graphic objects with the powerful support of Visual Studio.

#### 3. Microsoft Visual Studio

Microsoft Visual Studio is a IDE from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Besides, it can be applied on develop multi mobile platform application using an additional plug-in that is called Xamarin form. Since the proposed game will be developed using Unity3D, this IDE can have a lot of contribution on validating all the correctness of the syntax of C# language and providing some predicted codes with the technique of IntelliSense. Thus, it will speed up the development process.



4. Google Vision API (Protocol)

Google released this API with both cloud and mobile platform support. For cloud service, it enables image recognition features to identify most of the graphic image based on Google database server. For mobile platform, it can be used to detect faces, scan barcodes and recognize text even though the device is currently offline. By referring to the API documentations, developers can easily create some face detecting application without the knowledge of image processing since the API has covered everything from the complex algorithms.

5. Volley framework for Android platform (Protocol)

Volley is an open-source HTTP library that available on GitHub where it can ease to implement by converting data into JSON format before transfer data between Android applications and web services. Initially, Android platform required developer to have longer code for networking configuration. Fortunately, this framework allows developer to write less code for having the same outcome with some additional functions. For instance, it can automatically schedule network requests based on data transferring priority and support multiple concurrent network connections.

6. Microsoft Project

Microsoft Project is a project management software that allow us to create specific diagram such as Gantt chart that used to describe the duration and milestone for each respective task needed to complete the proposed system.

7. Visual Paradigm

Visual Paradigm is an ideal alternative diagramming software to Microsoft Visio. It able to create wide range of UML diagrams and flowchart that used to describe proposed system logical structure.

8. Evolus Pencil

Pencil is a lightweight open-source desktop GUI prototyping tool that developed by a group of Vietnamese where it uses to construct wireframes especially for mobile application and websites.

### 4.2.2 Hardware Requirements

#### 1. Laptop

Model	:	Acer Aspire R14
Processor	:	Intel® Core™ i5-5200U CPU @ 2.20GHz
RAM	:	12.0GB
System Type	:	64-bit Operating System
Graphics Card	:	NVidia GeForce 820m 2.0GB

#### 2. Mobile Device Specification

Model	:	Xiao Mi Max
Platform OS	:	Android
Android version	:	v6.0 (Marshmallow)
Processor	:	Hexa-core (1.4 GHz Cortex-A53 & 1.8 GHz Cortex-A72)
System Type	:	64-bit
RAM	:	3GB
Internal Memory	:	32GB
Camera	:	Primary 16 MP, Secondary 5 M

### 4.3 User requirements

Since agile development process is selected, user requirements are encouraged to write in user story format in order to figure out the problems effectively. There are some user requirements based on the proposed game that have been recorded within Table 4-1. Basically, these requirements are obtained from users though some conversations and all the desired features are considered implementable.

User stories
As a user, I want to control the whole game by using different kind of face expressions so that I can have better experience on playing it.
As a user, I want to use front camera for playing the game so that I able to play the game within a short distance without touching the mobile device.
As a user, I want to capture picture and record video automatically based on certain game conditions when I am playing the game so that I able to view and share my funny face expressions to social media.
As a user, I want to view the high score list of the game levels respectively so that I able to keep track and compete with the previous records.
As a user, I want the game contains a market store that able to provide some customizations for certain the game objects so that I can simply choose them based on my own preferences.
As a user, I want some social media services like Google Plus and Facebook that integrated within the game so that I can interact and compete the game high score with other people in social media while the game progress can be always backup as long as the device has network connection.
As a user, I want to have multi-player mode using technology of Bluetooth, NFC and Wi-Fi direct within the game so that I can interact with other users.

*Table 4-1: User Stories*

#### 4.4 System Performance Definition

Non-Functional Requirements	
1	<p><b>Description:</b> The game shall result in less memory consumption.</p> <p><b>Rationale:</b> The game shall be efficient use of memory and conserve the battery life for any kinds of devices.</p> <p><b>Fit criterion:</b> The game shall be able to read up to 90 percent of devices' specification automatically in current market in order to provide a suitable quality of game graphic so that the game performance will always be in the best condition.</p>
2	<p><b>Description:</b> The game shall emphasize on ease of use for all types of users.</p> <p><b>Rationale:</b> All the users shall think that the game is user-friendly and they can use it without using any external guide.</p> <p><b>Fit criterion:</b> The average acceptance rate of the game shall be more than 85 percent.</p>
3	<p><b>Description:</b> The game interface design shall be responsive across different devices.</p> <p><b>Rationale:</b> User shall be able to run the game across different devices regardless of devices' size and screen resolution issues.</p> <p><b>Fit criterion:</b> The average error rate of happening game graphics glitch shall be less than 2 percent.</p>
4	<p><b>Description:</b> The game shall be able to offer multinational language.</p> <p><b>Rationale:</b> This game should provide variety of selection for user to select a language that they are comfortable with.</p> <p><b>Fit criterion:</b> The game shall contain languages like English, Chinese, Japanese, Vietnamese and Malay if it is only released for Asia countries.</p>
5	<p><b>Description:</b> The game shall not display any regional sensitive sentences or message.</p> <p><b>Rationale:</b> User shall be comfortable when they playing the game and the game should not show any sensitive words.</p> <p><b>Fit criterion:</b> The rate of sensitive game contents shall be less than 0.5 percent.</p>

6	<p><b>Description:</b> The game shall be working with a minimum on 5 years old devices.</p> <p><b>Rationale:</b> The game shall support reverse compatibility on older devices for customers who own older devices to enjoy the game as well.</p> <p><b>Fit criterion:</b> The rate of device compatibility shall be exceeded 80 percent.</p>
7	<p><b>Description:</b> The game package shall include the terms &amp; condition and permission required.</p> <p><b>Rationale:</b> The game requires user's agreement to use in order to prevent abuses of game and protecting the game content to be stolen or reverse engineered by other parties.</p> <p><b>Fit criterion:</b> The terms &amp; condition and permission required shall be informed to users on the initial start-up and it will be disappeared once it has been agreed.</p>
8	<p><b>Description:</b> The game shall be able to respond within 5 seconds when it has been executed.</p> <p><b>Rationale:</b> The game shall be responsive to save time on waiting game response.</p> <p><b>Fit criterion:</b> The rate of issue occurring is less than 5 percent.</p>
9	<p><b>Description:</b> The game installation package shall be less than 30MB.</p> <p><b>Rationale:</b> The game shall optimize the usage of cloud computing and use less memory storage for user's initial installation.</p> <p><b>Fit criterion:</b> 80 percent of users are satisfied with it since it can be easily downloaded with a minimal network bandwidth.</p>
10	<p><b>Description:</b> The game shall provide social media account login.</p> <p><b>Rationale:</b> The game shall provide users an additional option of backup their game progress on cloud computing and interact and compete with other users from their social media applications respectively.</p> <p><b>Fit criterion:</b> The process of sign up, sign in, sign out and game progress synchronization shall not be exceeded 10 seconds.</p>

Table 4-2: Non-functional requirement

## **4.5 General Work Procedure**

### **4.5.1 Planning**

In initial stage, problem statements of the existing systems need to be defined come out with a proposed system to resolve the identified problems. Also, the project objectives and scope are defined to guide the development, to provide project overview and to predict the final output in end of development life cycle for stakeholders. Besides, all the relevant system specifications are gathered from stakeholder via meeting. The gathered specifications will be prioritized and chosen for current implementation. The development tools such as IDE for specific programming languages, database utility, wireframe tools and among other relevant software used in the project will be stated in the proposal.

### **4.5.2 Analysis**

Moreover, requirement gathering of the project is conducted to gather the information including user requirements and system requirements for the proposed system's specification and functionalities. A series of UML diagrams such as ERD diagram and Use Case diagram are developed to describe the relationship between each system module on how the proposed system functioning. Furthermore, a flowchart provides visual representation of steps and decision logic used to perform the processes of the proposed system. All analysis and requirements specification will be documented and undergo approval from relevant individual.

### **4.5.3 Design**

After the documentation being approved, the concept maps are designed to describe the overall system structure in a clear picture. Then, IDE like Visual Studio and Android Studio will be deployed by dragging and dropping some user controls into application forms in order to create the relevant proposed system blueprints. The blueprints are vital since the page contents and components are expected to be aligned in a proper manner for achieving better user experience. Once the blueprints are done, the first prototype will be created to gain feedback and approval from client and relevant personals.

#### 4.5.4 Implementation and Testing (Coding and Development)

Before the coding process is started, a few number of frameworks and libraries are encouraged to be configured for the project so that it can lighten and speed up the process of designing and theming the prototype and interface. After that, the functionalities of game development, including face landmarks movement detection will be started along with the game contents.

Since agile approach is chosen so that specification, development, and validation activities are interleaved rather than separate. Basically, there is a series version of game that used to be released and delivered to users for evaluation. Once feedbacks have been received, they will be filtered before planning for another update. Hence, the process cycle will be kept on repeating until reaching the ideal prototype and fulfilling majority of the requirements. This is considered the most time-consuming phrase among all the work procedures.

In addition, black-box testing approach should be implemented to test and verify the facial landmarks as input and how the game responses by comparing the proposed game's output against the other selected facial landmarks outputs. If the inputs entered will produced the desire output, then the proposed game is on the right track and reliable. Besides, using black-box testing can test the whole game from user point of view without having to know the complex programming logic behind the game. (Beal, 2015). Hence making testing easier since the main focus is on the input and desired output. Also, testers do not need to examine the programming code and how the program processes the output. Black-box testing is focusing on the inputs and outputs only without knowing the internal code implementation as shown as the figure 3-2. (Guru99, n.d.).



*Figure 4-2: Black-Box Testing Model*

#### 4.6 Project Timeline

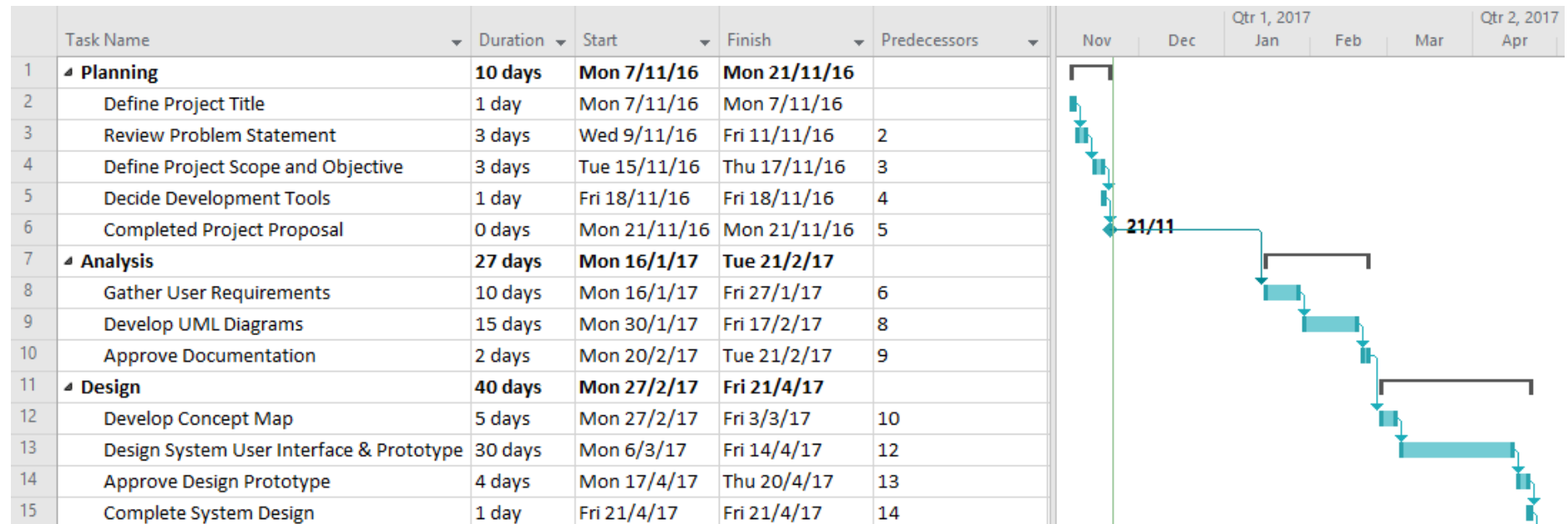


Figure 4-3: The Upper Half of the Project Gantt Chart



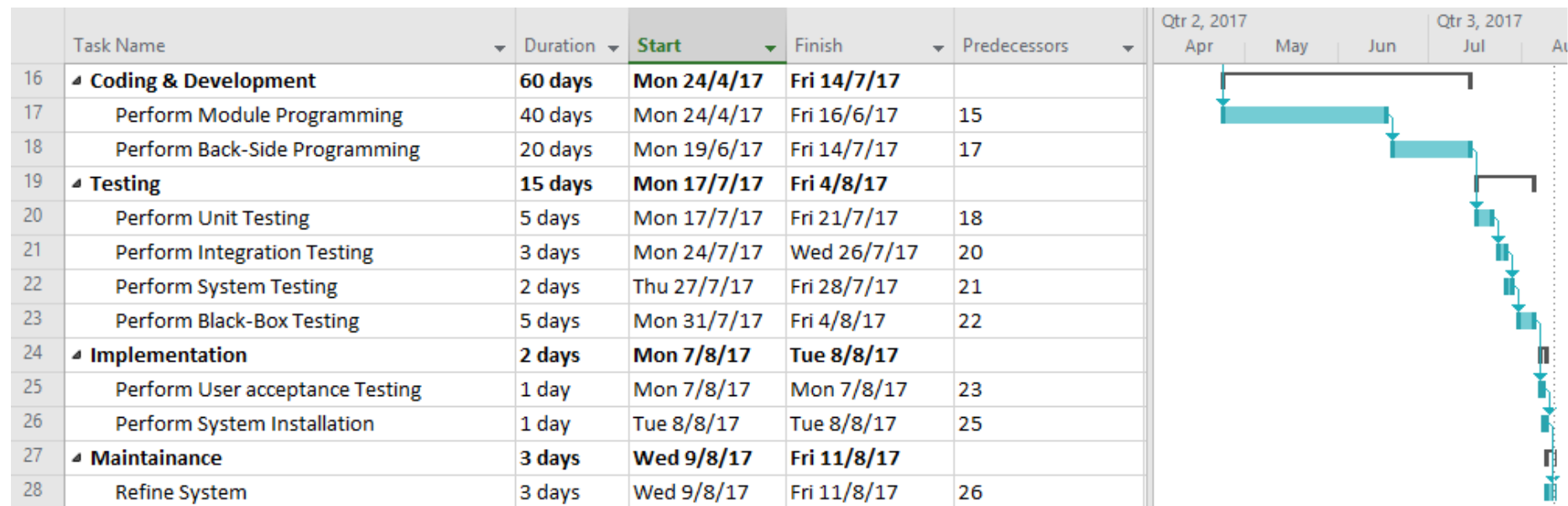


Figure 4-4: The Remaining Half of the Project Gantt Chart

According to figure 4-3 and figure 4-4, the overall durations to complete the entire proposed project are estimated to be 157 days.

## 4.7 Detailed Implementation

### 4.7.1 Step-by-step to compile the program

#### 4.7.1.1 Unity Game Development Engine

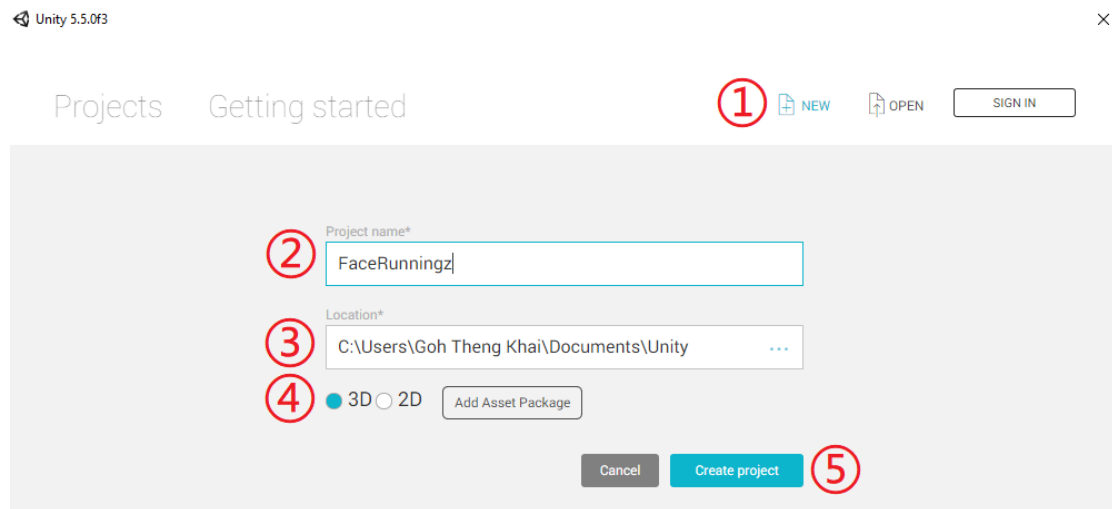


Figure 4-5: Unity Game Development Engine New Project Initialization Page

Firstly, click on new button to enter project initialization page. Then, key in project name and select project file path. Before click on Create project button, select type of project in term of 2D or 3D. After that, the new project will be created within a few seconds.

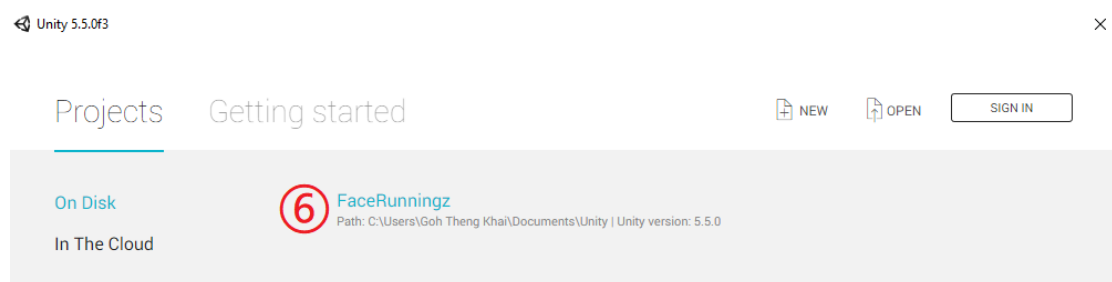


Figure 4-6: Unity Game Development Engine Project Selection Page

Secondly, select the project that has been created. This process will take couples of seconds to proceed.

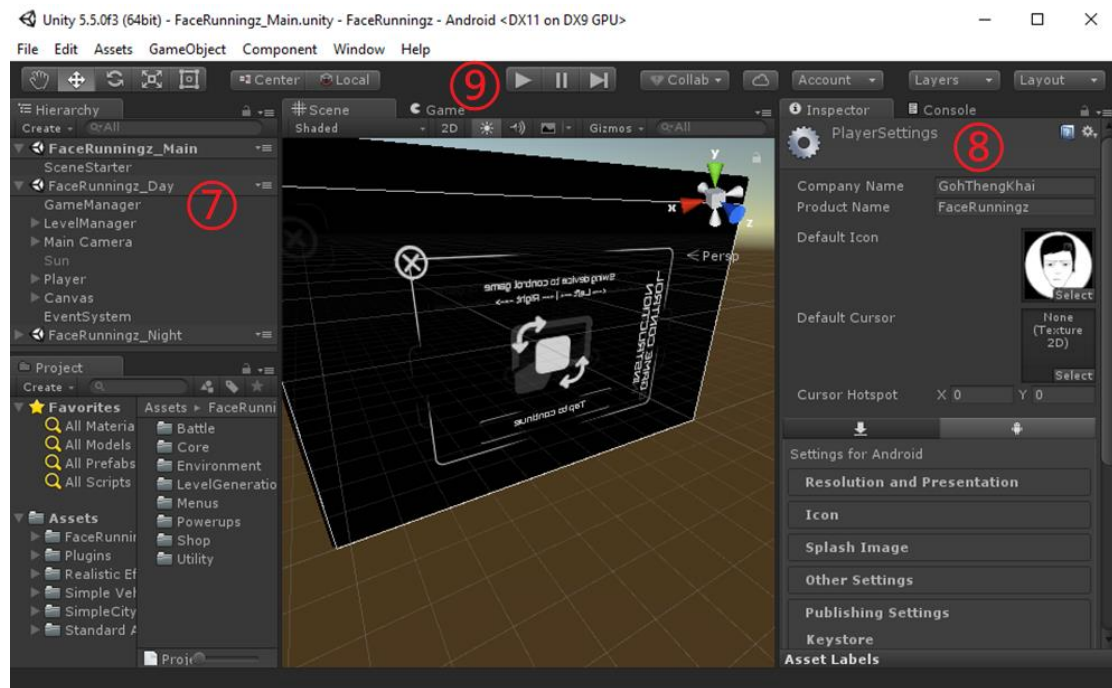


Figure 4-7: Unity Game Development Engine Main Interface

Thirdly, create or drag at least a game scene in hierarchy (label 7). Then, set up game details like game name, game icon, splash screen, installation path, screen orientation and others in Player Settings category (label 8). After that, click the play button (label 9) to test and debug the program and the debugging result will be displayed inside Console (above label 8).

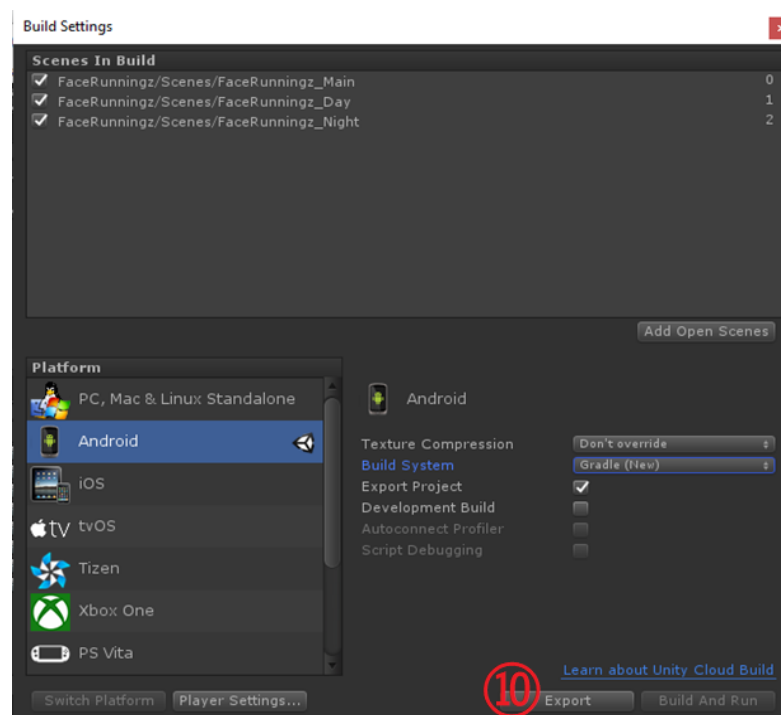
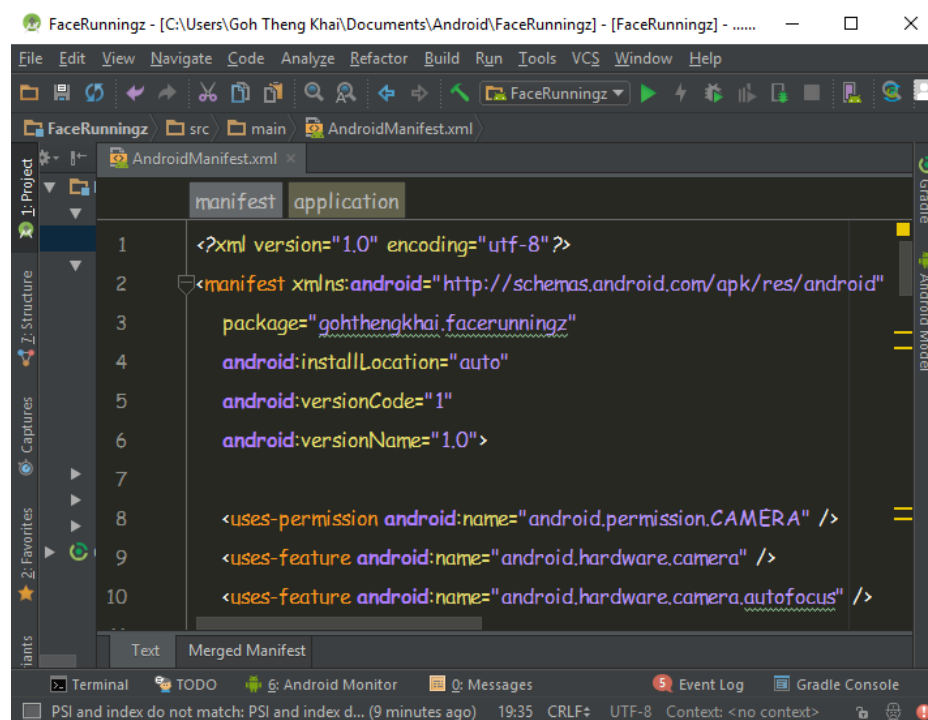


Figure 4-8: Unity Game Development Engine Build Settings

Lastly, click on File dropdown list and select build settings then figure 4-8 will be appeared. Once ensure the desired game scenes are selected, select appropriate platform before export the project to Android project that compatible with Gradle technique. Before proceeded to Android studio setup, merge the Unity project with the existing Android project.

### 4.7.1.2 Android Studio



*Figure 4-9: Android Studio Interface – Manifest Configuration*

Firstly, type in every single game details information, including hardware permission, preferred install location, appropriate theme, screen resolution, screen orientation, and the selected activity to be started.

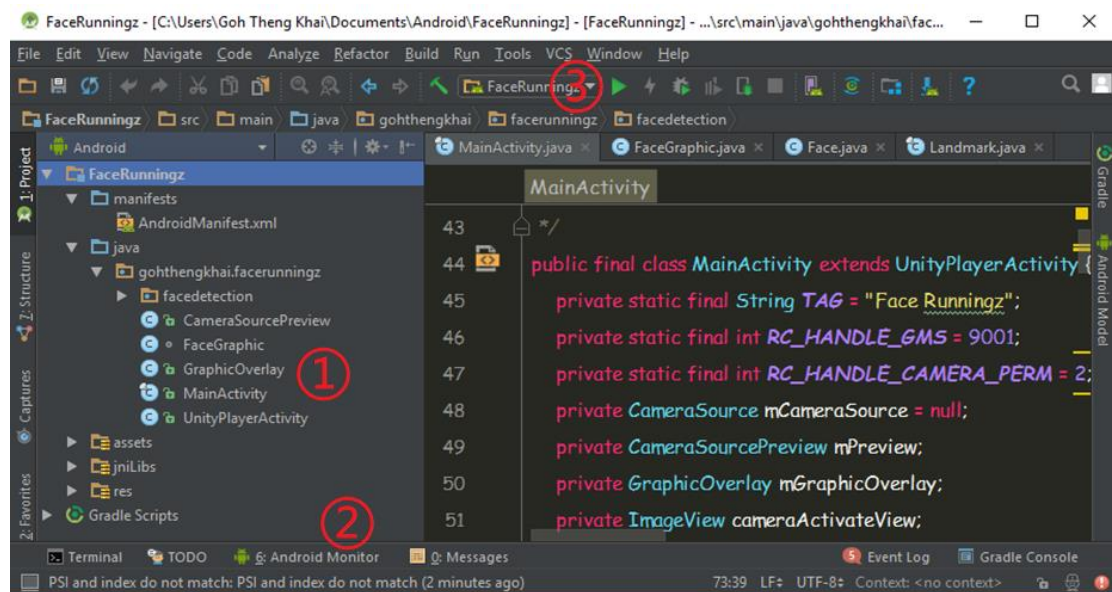


Figure 4-10: Android Studio Main Interface

Secondly, ensure relevant Java source code files in the left side (label 1). If want to activate debugging mode, click on Android Monitor (label 2) to prompt up a large text box otherwise this step can be ignored. After that, click on the play button (label 3) to trigger the debug process.

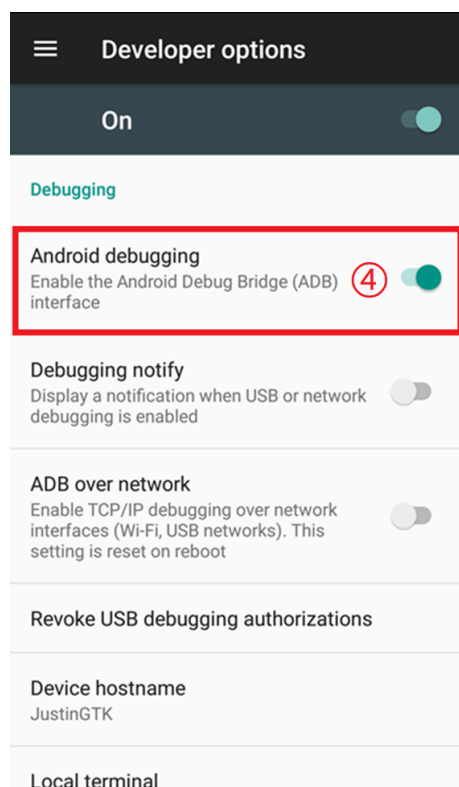


Figure 4-11: Developer option in Android device

Thirdly, ensure debugging mode has been enabled in Android device and connect the device with the personal computer via appropriate USB cable.

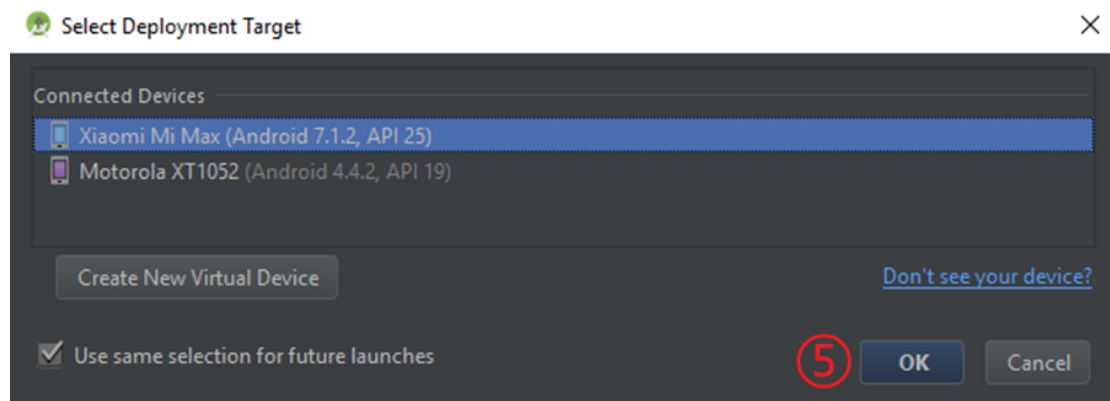


Figure 4-12: Android Studio Program Deployment Interface

Fourthly, select the plugged-in device model and click on OK button to start the debugging process. Basically, this process will take couples of minutes to compile for the first time. After that, the debugging process will be speeded up and usually finished within a minute. Finally, the program will automatically execute and appear on the plugged-in device.

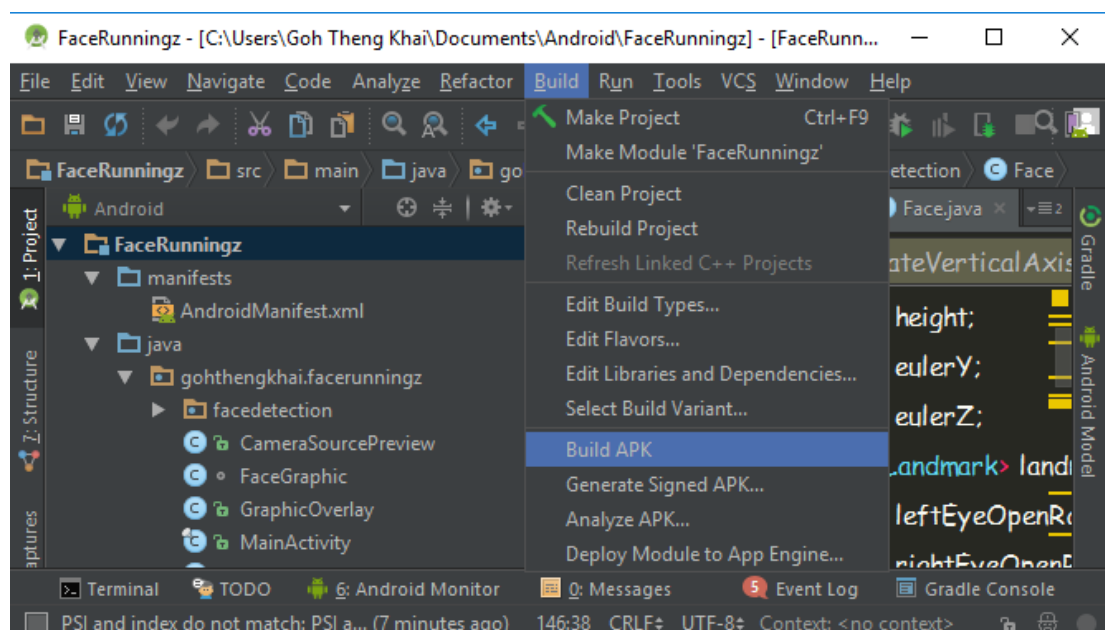


Figure 4-13: Android Studio Finalized Project Deployment Interface

If the project has been finished, click on Build dropdown list and select Build APK to generate the Android installation file so that the installer file can be published to Google Ply store, alternative stores or other websites.

### 4.7.2 Communication between Android Java and Unity C# Backend

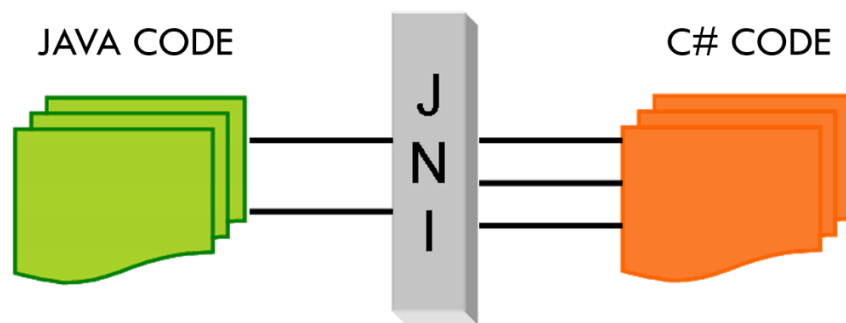


Figure 4-14: Visualized Communication between Java and C# languages

Java Native Interface (JNI) technique is required to interconnect between the backends of Java and C# source codes. Basically, all the proposed game interfaces are developed by using Unity game development engine with the help of C# backend scripting code. Unfortunately, Unity does not support face detection library so there are some alternative face detection APIs have been analysed and stated in literature reviews (chapter 2.6). At last, Google Mobile Vision Face API is selected because it is under Apache licence which is an open-source license and free to use. This API only supports on Android and iOS platform so it comes out with some difficulties. Fortunately, Unity support JNI communication technique by default but it has limited documentations so it is time-consuming to test and debug it.

Currently the proposed game is success to make interaction between Java and C# backends smoothly. There are two ways of communications, including Android to Unity and Unity to Android. Normally, Android to Unity communication is discouraged to use because it required more device resources and has worse performance whereas Unity to Android communication can perform better than other.

Android (Java) to Unity (C#) communication

```
UnityPlayer.UnitySendMessage("Unity_Game_Object_Name", "Method_Name",
"Parameters"); // Trigger Unity game object's method
```

Unity (C#) to Android (Java) communication (Unity Technologies, n.d.).

MainActivity.java contains: *public float HorizontalAxisValue = 0.0f;*

FaceController.cs contains: *public float GetHorizontalAxisValue() { return androidObject.Get<float>("HorizontalAxisValue"); } // Retrieve Android float value*

The above codes show that the concept and implementation of interaction between Android Java language backend and Unity C# language backend.



### 4.7.3 Game Scenes Concept and Description

The proposed game comes with 3 game scenes, including Simple City theme, Sci-Fi Blocks theme and scene selector. Scene selector is used to read the previous scene selection index and load it based on the sequence whereas both Simple City theme and Sci-Fi Blocks theme have their game scene respectively. For instance, Simple City theme obviously contains various types of buildings and a type of vehicle, including China Town entrance, Sky Tower, construction crane, flag, Statue of Liberty and SWAT Truck whereas Sci-Fi Blocks theme is built with the combination of rectangles and squares with a few of Sci-Fi textures.

Besides, the background scene that automatically move forward is based on infinite scrolling system. When user starts the game, he or she will think that the game object is moving forwardly but this is the false concept. Basically, the player's game object is stayed at the constant position from the beginning until the end of the game while the game contents of the world is kept on scrolling to left or right based on user input. Furthermore, the system will only generate the game objects that are under the player's camera view. Once user has passed through the game objects then they will be destroyed and removed automatically from the game scene for saving processing power and memory consumption.

In addition, the game system applied the stack data structure concept to manage its 2D menus. If user wishes to proceed to another menu, then the new menu will be stored and push into the stack. Once the user wishes to leave the menu then the system will pop up the current menu and use back the previous one. Thus, it is an efficient and effective way to keep track of all the menus regardless of the number of menus.

Lastly, the system contains a power-up system which allows to generate power-up objects randomly in the game world. Once player's game object has touched the objects then he or she can have additional special abilities for a few seconds while gaining some credits along the way. Currently, it only able to increase and decrease the movement speed of game object but it has potential to improve and enhance the module in future.



#### 4.7.4 Data Storing Technique – Shared Preferences

```

1  <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2  <map>
3      <string name="unity.player_sessionid">796218988333677811</string>
4      <int name="TotalCredits" value="83700" />
5      <string name="unity.cloud_userid">bd171f674c4184cafa120bbfac005260</string>
6      <int name="ShieldMode" value="0" />
7      <int name="DefaultThirdPersonCameraViewEnabled" value="1" />
8      <int name="RedFalcon" value="1" />
9      <string name="unity.player_session_background_time">1502800997828</string>
10     <int name="DefaultMobileInputMethod" value="1" />
11     <int name="SceneSelection" value="2" />
12     <int name="Screenmanager%20Is%20Fullscreen%20mode" value="0" />
13     <string name="ActiveSkin">RedFalcon</string>
14     <float name="RecordDistance133" value="434.0309" />
15     <float name="DefaultMusicAudioVolume" value="0.5" />
16     <int name="Screenmanager%20Resolution%20Width" value="1920" />
17     <string name="unity.player_session_elapsed_time">0</string>
18     <int name="BulletMode" value="0" />
19     <int name="Screenmanager%20Resolution%20Height" value="1080" />
20     <int name="__UNITY_PLAYERPREFS_VERSION__" value="1" />
21     <int name="GoldenSniper" value="1" />
22 </map>

```

Figure 4-15: Storing Method – Shared Preferences (XML)

Since the proposed game does not have too much of data to store, there is another way to store game data with lesser resources needed where it is known as shared preferences in Android. Basically, it is a XML file format that contains 5 primitive type variables, including Boolean, float, integer, long and string. Also, it only required specific name to identify its relevant data so it is easy to develop and manage. The shared preferences file usually cannot be accessed by user unless his or her Android device has granted with root permission.

In the proposed game, Shared preferences technique is selected and the storing sample is attached at figure 4-15. There is a game data protection included in the system to prevent root user to modify. If there is any modification from external sources then the data will be lost directly unless user abandons his or her current game progress by resetting the game data from device system.

```

<int name="TotalCredits" value="83700" />

<float name="RecordDistance133" value="434.0309" />

```

The above XML file format data are acquired from figure 4-15. For instance, the name with TotalCredits is the total amount of credits obtained by user whereas the name with RecordDistance133 is the high score of the game scene 1 (Simple City) with input selection 3 (Sensor mode) in game mode 3 (Hard mode).

#### 4.7.5 High-Fidelity Game Prototypes

The below figures are almost same as the actual program interface. Besides, the proposed game can be downloaded via link: <<https://goo.gl/ni6xr2>> where it only compatibles with Android 5.0 and above.

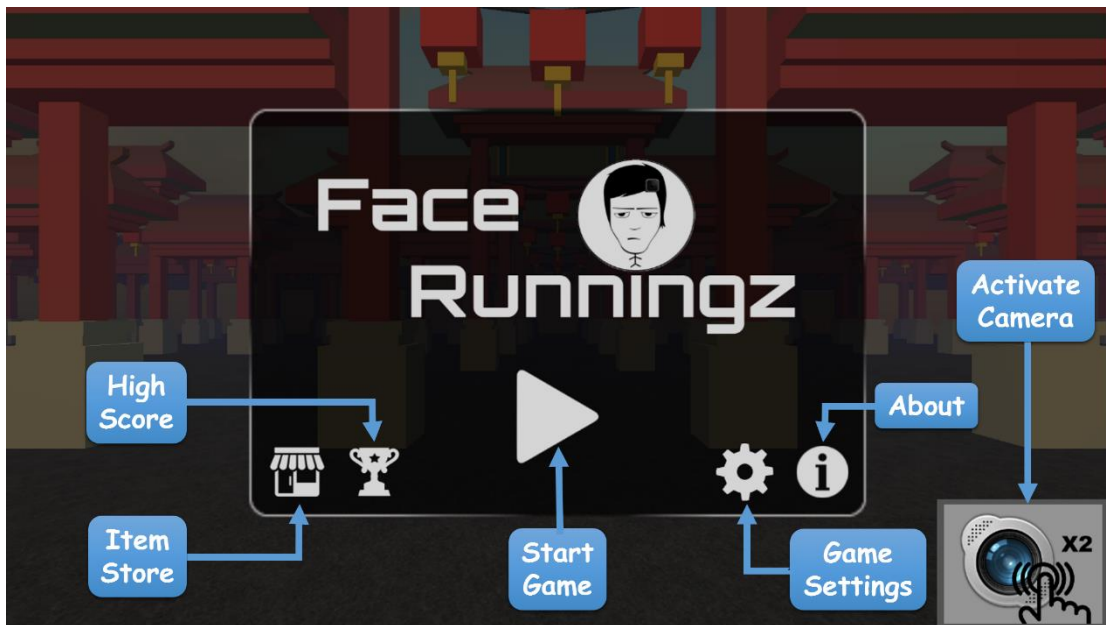


Figure 4-16: Main Menu Interface

When entered the game, the system will display a splash screen before it proceeds to main menu interface. After clicked Start Game button, then Game Option menu (figure 4-17) will be appeared. Then, select game scene and level to proceed further.



Figure 4-17: Game Option Selection 1 Interface



*Figure 4-18: Game Option Selection 2 Interface*

If user choose to select game scene then another game scene (figure 4-18) will be loaded and the game background will be changed. After that, game instruction interface will be prompted up. User is required to activate the camera since the game required user to smile for proceeding further game operation. Once the camera has been activated, the system will read the facial information from user's face (figure 4-19). For instance, X:0 and Y:0 represents face is exactly straight, X:-1 represents left face and X:1 represents right face whereas Y:-1 represents bottom face and Y:1 represents top face.



*Figure 4-19: Game Instruction Gameplay Interface*



Figure 4-20: Gameplay 1 Interface

User can turn his or her face to left or right side for moving the spaceship so that it can avoid from crashing the buildings. While user is playing, the system will record the distance obtained. If someone breaks the high score, then the system will prompt up a “New Record” message for a few seconds before it disappeared. If spaceship touched the power-up object, then extra credits will be given to user. Besides, user can choose to pause the game by clicking Pause button anytime, then pause menu (figure 4-21) will be prompted up. User can either resume game or give up by clicking relevant buttons.



Figure 4-21: Pause Menu Interface





Figure 4-22: Restart Menu 1 Interface

If user collided with the obstacle, then restart menu will be prompted up. If user wants to restart the game, he or she can click on Restart button or smile to proceed the operation. User can enter item store quickly without the need of back to main menu. If user wants to leave the game, he or she can click on Back button to back to main menu. Moreover, if user breaks the high score, the system will show “New Record!” (figure 4-22) message otherwise it will display distance travelled, relevant high score and credits obtained information only.



Figure 4-23: Restart Menu 2 Interface

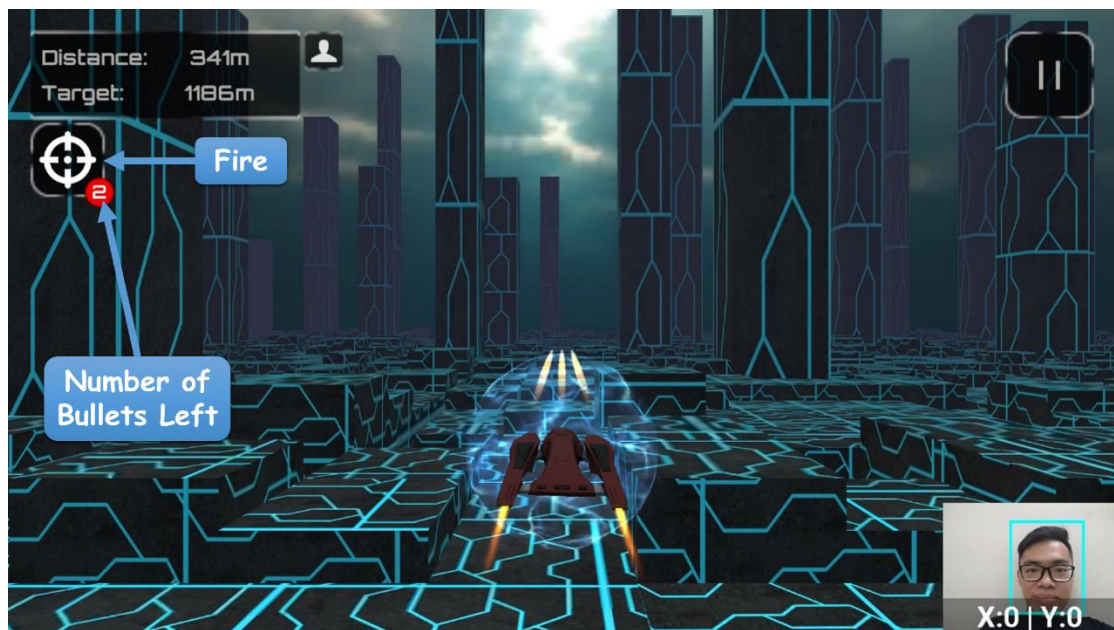


Figure 4-24: Gameplay 2 Interface

If the shooting mode is activated, then user allows either to click on Fire button or to smile for destroying the obstacles. Normally, it has limited of bullets to use and the number of bullets will be deducted by 1 if a bullet is consumed. In addition, if shield mode is activated, then there will be a force shield surrounded the spaceship and it gives an additional chance to play within a round. These modes will be activated and consumed directly after entered the game so they will still disappear after play for a round even their effects cannot be not finished.

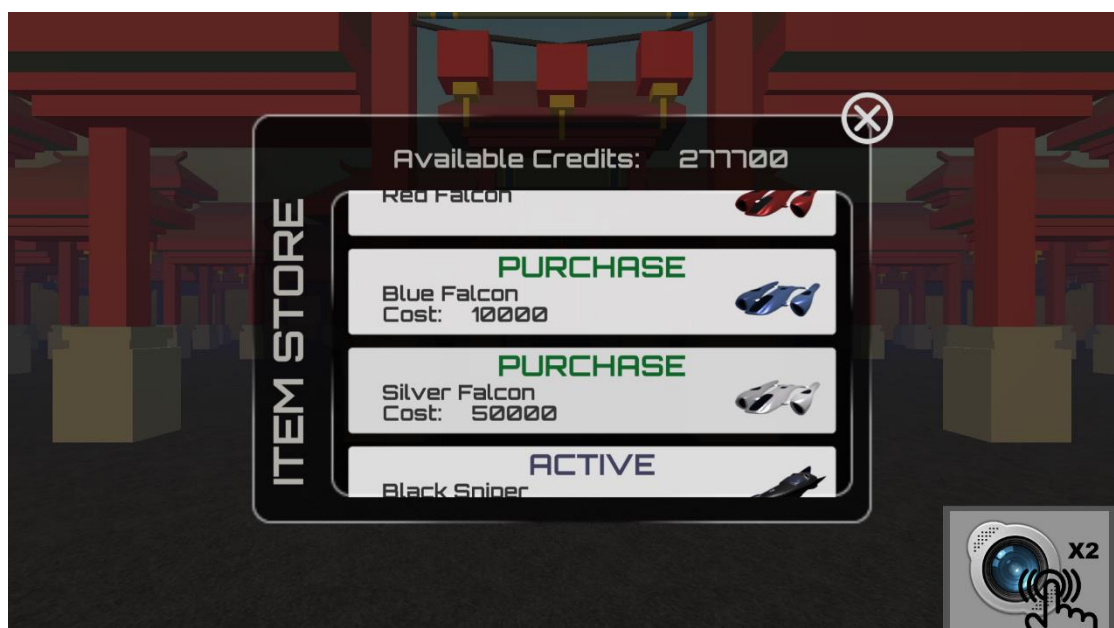


Figure 4-25: Item Store Interface 1



Figure 4-26: Item Store Interface 2

Based on figure 4-25 and figure 4-26, the system allows user to switch the player's game objects and unlock certain modes, including credits multiplier, shooting and shield modes. If user has sufficiency of credits, he or she can purchase it and the credits will be deducted, otherwise there an error sound effect will be played to alert user. Meanwhile, figure 4-27 shows that high score board interface that able to display various types of high score. User also allows to view other high score by clicking on the display selection buttons then relevant high scores will be updated to the interface.



Figure 4-27: High Score Board Interface



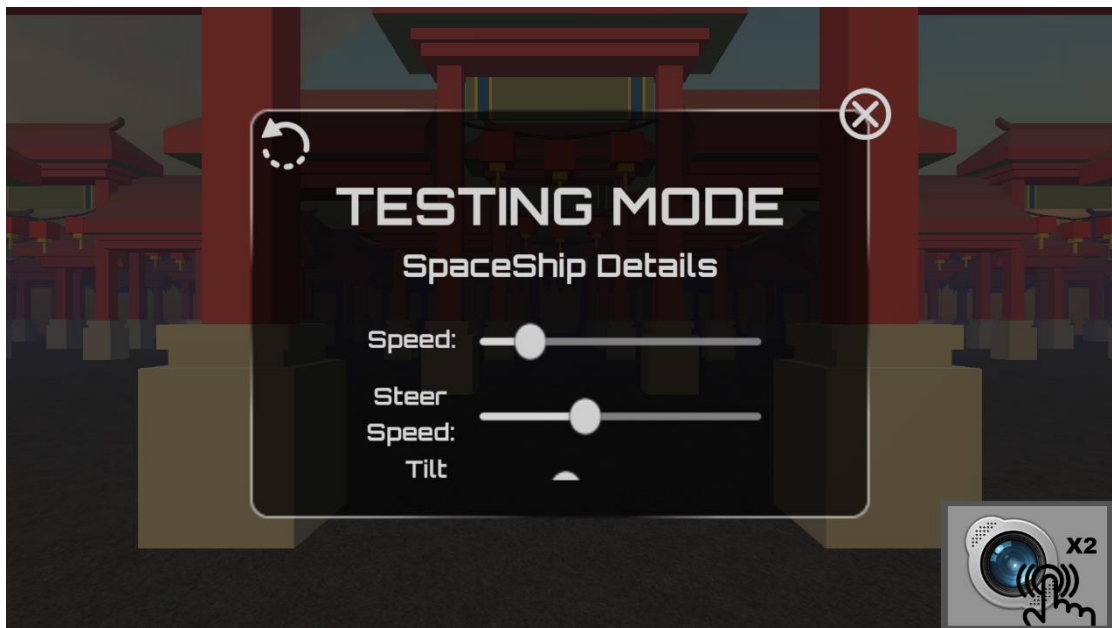
*Figure 4-28: Game Settings Interface*

Figure 4-28 shows that game settings interface that allows user to change playing method in term of face mode, touch mode and sensor mode, to change camera view in term of third-person view or first-person view, and to adjust the volume of background music and sound effect. According to figure 4-29, it is a game about interface that displays developer details but it contains a hidden testing mode. This mode can be activated by clicking UTAR icon and its purpose is to perform testing operation by developer.



*Figure 4-29: Game About Interface*





*Figure 4-30: Testing Mode Interface 1*

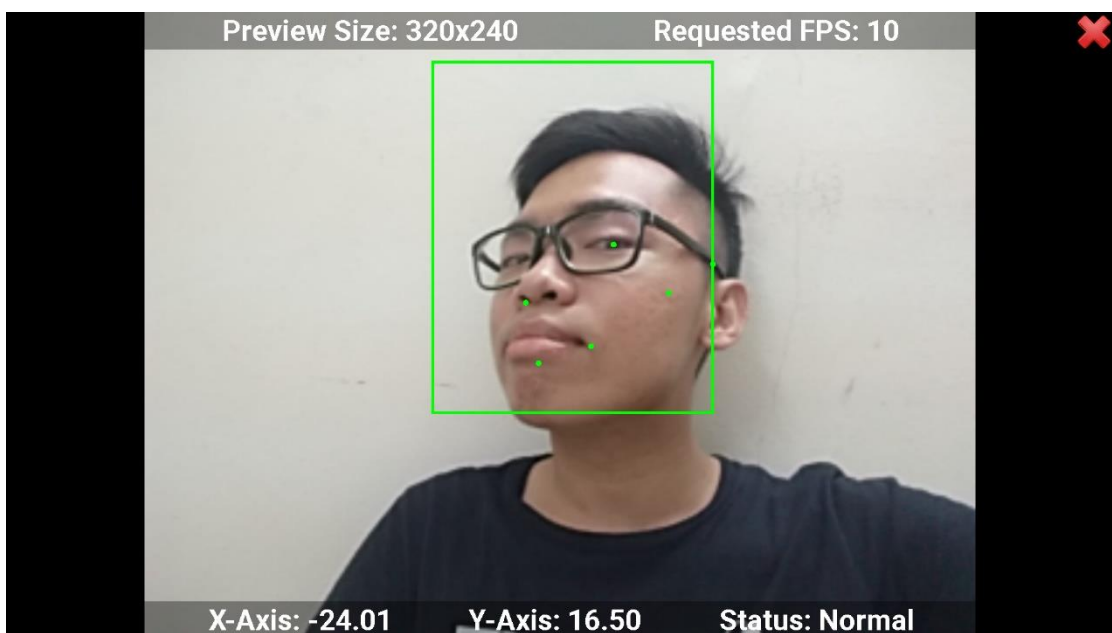


*Figure 4-31: Testing Mode Interface 2*

Based on figure 4-30, figure 4-31 and figure 4-32, they are components within testing mode. Developer allows to adjust spaceship movement properties, including speed, steer speed, tilt angle and tilt speed, to enable height motion mode for up and down movement purpose, to enable god mode for performance and game scene testing purpose as the game object will not be collided with any obstacle. Lastly, there is a coin editor attached inside testing mode so developer can use it to test the actual objects' behaviours of item store. This mode will be removed once the game reveals to public.

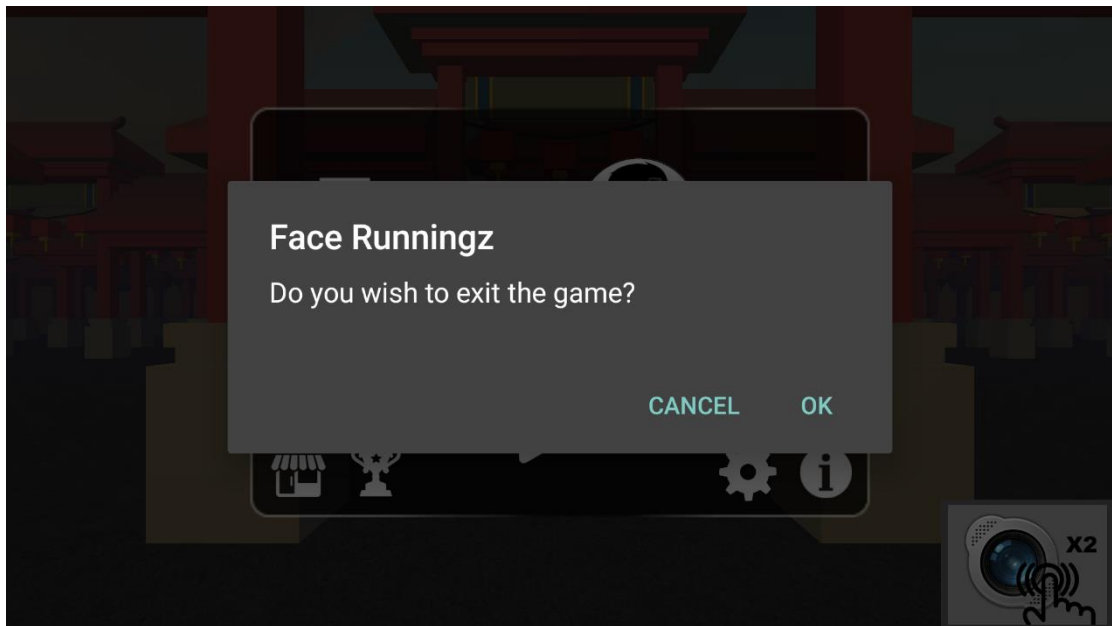


*Figure 4-32: Testing Mode Interface 3*



*Figure 4-33: Enlarged Camera Interface*

There is a hidden option on the camera icon. Once the camera has been activated, the enlarged camera interface can be accessed by long tapping the camera. Basically, there will display some camera and facial information like preview size, requested FPS rate for face detection, face position in term of x-axis and y-axis, smiling rate, and the probabilities of winking and blinking eyes. Thus, the additional information can be developed further as it still has quite some expansion spaces. For instance, winking or blinking eyes to perform other game operations.



*Figure 4-34: Program Exit Interface*

Once Android device native back button has been tapped, the dialog box (figure 4-34) will be appeared directly. Once user clicks on OK selection, the game will be terminated otherwise nothing will be happened.

#### 4.8 Verification Plan and Program Testing

The verification plan of the proposed game is followed the method of black-box testing model as it displayed on the figure 4-2. Basically, it required developers to figure out any possibility of inputs and come out with its expected output. After the game has been programmed, compare both expected output and actual output to determine the level of project completion.

Launch Game Application		
Input	Expected Output	Actual Output
Click on game icon	Display animated splash screen.	Displayed.
	Main menu interface will be displayed within 5 seconds.	Displayed but it takes around 6 to 7 seconds.
	Request related hardware permissions from user on the initial start-up.	Displayed.

*Table 4-3: Game Launch Verification Plan*

Game Main Menu Layout		
Input	Expected Output	Actual Output
Click on Single Player button	Switch from main menu layout to single player layout.	Displayed.
Click on Multi-player button	Switch from main menu layout to multi-player layout.	Null since the function is yet to be implemented.
Click on About button	Display the information of developer and game details.	Displayed.
Click on Market Store button	Switch from main menu layout to item store layout.	Displayed.
Click on Leader Board button	Switch from main menu layout to high score layout.	Displayed.
Click on Options button	Switch from main menu layout to game configuration layout.	Displayed.

*Table 4-4: Game Main Menu Verification Plan*

Game Market Store Layout		
Input	Expected Output	Actual Output
None	Display current amount of coins.	Displayed.
Click on More Coin button	Display the list of coin purchasing options along with different price respectively.	Displayed.
Click on Purchase Button	Prompt a success purchased message if current coin amount is more than coin required and coin amount will be deducted.	Displayed.
	Prompt an error purchasing message if current coin amount is insufficiency.	Played an error sound effect instead of prompt up error message.
Click on 'Purchased' button	Nothing will be happened since button has been disabled. It shows that clicked item has been purchased previously.	Displayed.
Click on another category tab button	Shopping list menu layout will be switched to another item layout.	Failed to display as item store only contains a list of items to be purchased.

Table 4-5: Game Market Store Verification Plan

Single Player Game Configuration Layout		
Input	Expected Output	Actual Output
Click on Game Mode button	Switch from configuration layout to gameplay layout.	Displayed.
Click on More Info button	Pop up a dialog box for gameplay guideline.	Displayed but on different game displayed sequence.

Table 4-6: Single Player Configuration Verification Plan

Multi-player Game Configuration Layout (Function is yet to be implemented)		
Input	Expected Output	Actual Output
Click on connectivity selection drop-down list	Display a list of connectivity options like Bluetooth, Wi-Fi direct and NFC based on the specification of device.	Null.
	Hardware request will be prompt out in order to proceed further.	Null.
Click on Refresh button	Update the list of possible devices to play.	Null.

*Table 4-7: Multi-player Configuration Verification Plan*

Game Configuration Layout		
Input	Expected Output	Actual Output
Drag scale controller on music and sound effect	Game music and sound effect will be changed according the position of the scale controller. (Drag to right side for increasing volume and vice versa)	Displayed.
Click on Social Media button	Pop out another account layout for sign in, sign out and game progress synchronization.	Null since the function is yet to be implemented.

*Table 4-8: Game Configuration Verification Plan*

Gameplay Layout		
Input	Expected Output	Actual Output
Click on the Start Game button	Camera preview is started.	Displayed.
	Display game environment.	Displayed.
	Display game detail such as remaining time, remaining life, coin acquired and current score.	Displayed.
Detect face using front camera	Prompt a success message if a face is detected. If there are more a face is detected, then the first face will be taken as the game controller.	Slightly different output. It does not prompt up success message and it will only detect one of the nearest human face and the retrieval information is able to be the game controller.
Read the angles of facial landmarks from first face	Game object like ball or spaceship will be moved further and the direction of game object will be changed according to the angles of facial landmark movement.	Displayed but the smoothness of the game is depended on devices' specifications.
Read face expression	Game object like ball or spaceship will be perform some additional game operations like to jump or to destroy obstacles.	Displayed.

Table 4-9: Gameplay Verification Plan

After proceeded the verification, the proposed game is most slightly accepted as the actual program has passed and reached more than 70% of the expected outputs.

## **Chapter 5: Conclusion**

### **5.1 Project Achievements**

Based on verification plan and testing operation that attached in chapter 4.8, actual outputs of the proposed game are quite satisfied as it reached almost all the expected outputs and fulfilled all the project objectives. Basically, the game can detect user face movement and expression for controlling the game object with direction of left, right, up and down and triggering the game object to shoot and destroy other object obstacles. Besides, game item store able to provide several options for user to make his or her own customization to the game object and a few of extra game modes which required to certain amount of game credits to unlock. Moreover, there are currently a couple of different game scenes which allows user to switch and each game scene provides 3 game levels, including easy mode, normal mode and hard mode while their high scores will be stored respectively so that the game will be more challenging to user. Furthermore, the game also allowed user to adjust audio volume, select different playing methods and change the object view in term of third-person or first-person view.

### **5.2 Project Reviews**

#### **5.2.1 Project Strengths**

The proposed game uses front camera to interact between user and device so user can play the game through his or her face movement without touching the device after it has been setup. This idea is quite new and novelty as minority of similar games can be found in current application market store. Besides, the game is quite flexible as it provides 3 types of input playing methods so user can select the most suitable method to play. Overall the game is simple to play without guideline, user can adapt to it within a few minutes.

#### **5.2.2 Project Limitations**

Although the game is able to detect user face movement but it does not really stable for up and down direction since the selected face detection API does not have this features by default. Moreover, it does not have social media integration like login and bind the game progress and high score to Facebook and Google+ so the game data will be lost easily if user accidentally clears the application data or switches to a new mobile device. In addition, the game may lack of attractiveness since the number of game scene and



items from item store are quite limited. Lastly, the device that does not come with prebuilt Google Framework is unable to activate face mode as the selected face API is fully depended on that whereas the rest of game modes will be functioned as expected. This issue mostly likely will not be happened except country like China that restricted Google services.

### **5.3 Implementation Issues and Challenges**

In this project, there are several technical challenges that encountered along the way. First of all, there are some development tools that required to figure out before the project can proceed further. Basically, there are some integrated development environments (IDE) like Eclipse, NetBeans, Android Studio and Visual Studio that needed to be considered. As Eclipse IDE and NetBeans IDE are required some additional plugins for developing Android applications and these plugins are no longer providing supports by Google Inc since year 2016, Android Studio is considered the best choice among all IDE for Android platform.

Moreover, in order to construct a 3D game graphic environment, Unity game engine is required and considered the most appropriate tool where some graphic resources have been provided in its market store and developer can create the game environment by dragging and dropping these graphic components just like playing a puzzle game. Nevertheless, most of better graphic components are required to purchase in order to adopt them while most of free components are quite common and perhaps they do not have good-looking and high-quality graphic. Also, some behaviours of game components may need to re-modify where it required certain knowledge level of scripting programming languages like C# and JavaScript. Thus, time-consuming is necessary in order to seek for nicer components, to learn all these relevant programming languages and to adapt to the user interface and layout of Unity game engine.

Furthermore, the proposed game is planned to develop on Android mobile platform. As the fundamental of Android development is based on Java programming language, it will require a strong based of Java knowledge in order to handle any kind of features implementations and coding exceptions within the minimal of time for ensuring project will always be conducted efficiency and effectively. Although most of static layouts can be constructed using drag and drop option but it will still involve some dynamic layouts where there are thousands of coding patterns regarding to user interfaces but it

required developer to keep on updating its coding keywords as those keywords may be abandoned or replaced with other keywords. Game layouts that are created from Unity game engine are required to interact within Android Studio as it will take some efforts to set up some configurations as it is unfamiliar. Hence, it will lead to some difficulties to developer.

In addition, the project is required to perform face detection so mobile computer vision library is needed. Google Vision API is considered the best options to implement as it is under Google product and Apache licence so it should be fully compatible with Android Studio and free to use. Basically, it is a light-weight Java based library that allows camera to detect facial landmarks without internet where it supports the development on both Android and iOS platform. It is also convenience to apply as developer does not require any knowledge of face detection algorithms since the API has covered almost everything. Nevertheless, the compiled application is unable to execute in Android devices which do not have integrated Google Play framework so some segments of users may not be benefited from it especially China citizens. Besides, there are less samples and solutions of the library available on the internet as it is unfamiliar and unpopular. Also, there are insufficiency of API documentation within the library as it will gain extra workloads for developer as he or she may need to perform a lot of trial-and-error methods before the actual project can be implemented.

Besides, the communication between Android Java and Unity C# (chapter 4.7.2) backends is quite complex to handle since both programming languages have different syntaxes. After several researches have been made, the communication finally can be established via JNI technique that provided by Unity game engine. Indirectly, it makes the program compilation to be more complicated since developer required to export the Android type project and merge it with the existing Android project before the final APK installer can be done. These steps will take around 5 minutes to complete.

Consequently, it is hard to find on a proper or complete related game sample from the existing application market like Google Play. There are only a few samples that are available online but most of them are just under personal development where other users do not have opportunity to test it as those developers only share their ideas and applications within videos through internet. Hence, developer is hardly to take these samples as references as all the coding parts have been hidden.

### 5.4 Future Work

Firstly, social media integration should be included so user can login to Facebook or Google through his or her own account to keep track and ensure the safety of the game progress. If the game data is lost, then user can perform synchronization in order to retrieve back his or her personal progress within seconds.

Secondly, the game should increase the number of game scene and game objects in item store to attract more users to play. Also, the game should raise its complexity in term of contents. For instance, the game provides daily and weekly rewards to user who able to play at the appropriate time.

Thirdly, once the game contains more contents then achievement system can be implemented. It can use to keep track of some specific records like playing duration, unlocked certain spaceship and or reached certain distance. This achievement system usually can be done with the help of API. For instance, Google Play Game API already provides the features and it able to link to social media account. Thus, the game will be attracted by some users as they will try their best to unlock these achievements and gains some experience for their game account.

Fourthly, in-app purchase option should be implemented for user who is willing to perform better game experiences while the profit can use to support developer. It is considered the latest trend for majority of games in the market.

Fifthly, the game menu layouts can be enhanced by changing their size, brighter colour, animated effects, and shape for better appearance and user attraction. For instance, 3D menu interfaces can be applied to the game for better looks.

Sixthly, boss mode can be implemented into the game for additional modes. Basically, the behaviour of boss can be integrated with the technique of artificial intelligence (AI) so the way of attacking the user will be performed dynamically.

Lastly, multiplayer mode should be applied into the game for enhancing interaction among users. Basically, the game can provide extra modes like surviving and timing mode for users to make competition. The connection between device and another device can be done through Bluetooth, NFC or online. Other than above suggestions, there are still plenty of ideas and implementations for better game enhancement.

## 5.5 Summary

First of all, users might be getting bored on most of the existing games in current market. Although the game contents are always different from each other but the ways of interacting or controlling the game are always the same as they are fully depended on common input devices. As the result, motion-control game is the most suitable choice to handle it. Unfortunately, most of the related games are required specific hardware where they are quite expensive to certain users.

In order to handle these issues, the proposed project is aimed to develop a motion-control mobile game by focusing on low hardware requirement so that every single user is able to experience it with no cost. Nowadays, most of the low-end and mid-range Android devices already contain at least an average-quality front camera and the proposed game is totally fulfilled with this condition. Basically, the project scope is to ensure that every single game operation like clicking a button should only be involved using facial landmarks recognition technique on real-time camera so that user can play the game without touching the device for a specific distance.

At the end of this project, it will bring out a complete program as user can play the game using facial landmarks detection in order to control some game objects in a 3D game environment using Unity3D game engine. If user's face expressions have met certain condition, then the game objects will perform additional behaviors like shooting and restart the game. Due to the time constraint, there are tons of improvements and new features that stated in chapter 5.4 will be continued to develop for making it robust and competitive in the upcoming future.

## Bibliography

- Beal, V. (2015). *What is Black Box Testing?*. [Online] Webopedia. Available from: <[http://www.webopedia.com/TERM/B/Black\\_Box\\_Testing.html](http://www.webopedia.com/TERM/B/Black_Box_Testing.html)>. [21 November 2016].
- Cabebe, J., & Mitroff, S. (2013). *Snapchat's latest updates bring video chatting and instant messages to this secret sharing app*. [Online] Available from: <<https://www.cnet.com/products/snapchat-ios/review/>>. [24 November 2016].
- Clatworthy, D. (2016). *Tech review: Facerig*. [Online] Available from: <<http://gamers-haven.org/haven-review-facerig/>>. [20 November 2016].
- Facebook Inc. (2016). *MSQRD - Android Apps on Google play*. [Online] Available from: <<https://play.google.com/store/apps/details?id=me.msqrd.android>>. [24 November 2016].
- FaceRig. (2015). *FaceRig on steam*. [Online] Available from: <<http://store.steampowered.com/app/274920/>>. [20 November 2016].
- Google Cloud. (2016). *Vision API - image content analysis / Google cloud platform*. [Online] Available from: <<https://cloud.google.com/vision/>>. [19 November 2016].
- Google Mobile Vision. (2015). *Face Detection Concepts Overview / Mobile Vision / Google Developers*. [Online] Available from: <<https://developers.google.com/vision/face-detection-concepts>>. [7 March 2017].
- Guru99. (n.d.). *What is Black Box Testing?*. [Online] Available from: <<http://www.guru99.com/black-box-testing.html>>. [21 November 2016].
- Holotech Studios. (2016). *FaceRig - Android Apps on Google play*. [Online] Available from: <<https://play.google.com/store/apps/details?id=com.holotech.facerig>>. [20 November 2016].
- Indiegogo. (2014). *FaceRig*. [Online] Available from: <<https://www.indiegogo.com/projects/facerig#/>>. [20 November 2016].
- Luxand. (2016). *Luxand - Detect Human Faces and Facial Features*. [Online] Available from: <<https://www.luxand.com/facial-features>>. [7 March 2017].

## Bibliography

- Luxand. (2017). *Detect and Recognize Faces and Facial Features with Luxand FaceSDK*. [Online] Available from: <<https://www.luxand.com/facesdk/>>. [7 March 2017].
- Microsoft Cognitive Service. (2016). *Face API* [Online]. Available from: <<https://www.microsoft.com/cognitive-services/en-us/face-api/>>. [7 March 2017].
- Smith, J. (2016). *MSQRD App details: Live filters for Selfies & videos*. [Online] Available from: <<http://www.gottabemobile.com/2016/03/09/msqrd-app-live-filters-for-selfies-videos/>>. [24 November 2016].
- Snap Inc. (2016). *Snapchat - Android Apps on Google play*. [Online] Available from: <<https://play.google.com/store/apps/details?id=com.snapchat.android>>. [24 November 2016].
- Unity Technologies. (n.d.). *AndroidJavaObject*. [Online] Available from: <<https://docs.unity3d.com/ScriptReference/AndroidJavaObject.html>>. [18 August 2017].

# **Attachment (CD)**

<b>Universiti Tunku Abdul Rahman</b>			
<b>Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION  
TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	Goh Theng Khai
<b>ID Number(s)</b>	14ACB00092
<b>Programme / Course</b>	Information Systems Engineering (IA)
<b>Title of Final Year Project</b>	Development of Interactivity Game using Facial Landmarks

<b>Similarity</b>	<b>Supervisor's Comments</b> (Compulsory if parameters of originality exceeds the limits approved by UTAR)
<b>Overall similarity index: <u>1</u> %</b>  <b>Similarity by source</b> Internet Sources: <u>1</u> % Publications: <u>0</u> % Student Papers: <u>-</u> %	
<b>Number of individual sources listed of more than 3% similarity: <u>0</u></b>	
<b>Parameters of originality required and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

\_\_\_\_\_  
Signature of Supervisor

Name: \_\_\_\_\_

Date: \_\_\_\_\_

\_\_\_\_\_  
Signature of Co-Supervisor

Name: \_\_\_\_\_

Date: \_\_\_\_\_



## Chapter 1: Introduction

### 1.1 Project Background

Face recognition is a computer technique that has ability to identify or recognize a person based on the characteristics of his or her face. Normally, a digital image or a video frame from a video that contains human face will be used to analyse and extract its meaningful features and characteristics in terms of inner features and outer features such as eyes, nose, mouth, head shape and hairline. After that, comparing and matching them with a facial database then putting them into a useful representation and performing some kind of classification on them.

Basically, face recognition is considered quite an easy task for a human being to identify known faces. Nevertheless, human being has limited memory and not skilful on dealing with a large amount of unknown faces. Major computers in the existing current market

### Match Overview

1%

<		>
1	<a href="#">alternativeto.net</a> Internet Source	<1% >
2	<a href="#">ilri.org</a> Internet Source	<1% >
3	<a href="#">taw.net</a> Internet Source	<1% >
4	<a href="#">dspace.cc.tut.fi</a> Internet Source	<1% >
5	<a href="#">conservancy.umn.edu</a> Internet Source	<1% >
6	<a href="#">dspace.thapar.edu:8080</a>	<1% >

Turnitin Result Figure 1

Turnitin Result Figure 2 and Turnitin Result Figure 3

# FYP2\_FacialLandmarksGame

by Goh Theng Khai

Submission date: 21-Aug-2017 06:40PM (UTC+0800)

Submission ID: 838611972

File name: FYP2\_FacialLandmarksGame.pdf (5.34M)

Word count: 16566

Character count: 86475

## FYP2\_FacialLandmarksGame

### ORIGINALITY REPORT

1 %

SIMILARITY INDEX

1 %

INTERNET SOURCES

0 %

PUBLICATIONS

%

STUDENT PAPERS

### PRIMARY SOURCES

1

[alternativeto.net](#)

Internet Source

<1 %

2

[ilri.org](#)

Internet Source

<1 %

3

[taw.net](#)

Internet Source

<1 %

4

[dspace.cc.tut.fi](#)

Internet Source

<1 %

5

[conservancy.umn.edu](#)

Internet Source

<1 %

6

[dspace.thapar.edu:8080](#)

Internet Source

<1 %

7

[conferences.ulbsibiu.ro](#)

Internet Source

<1 %

Exclude quotes Off

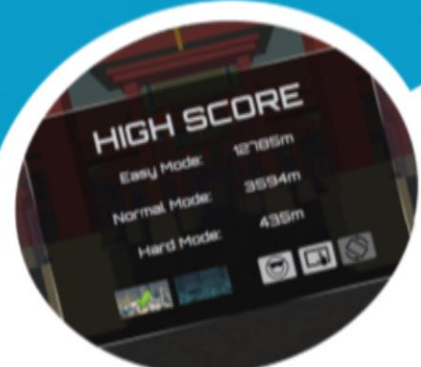
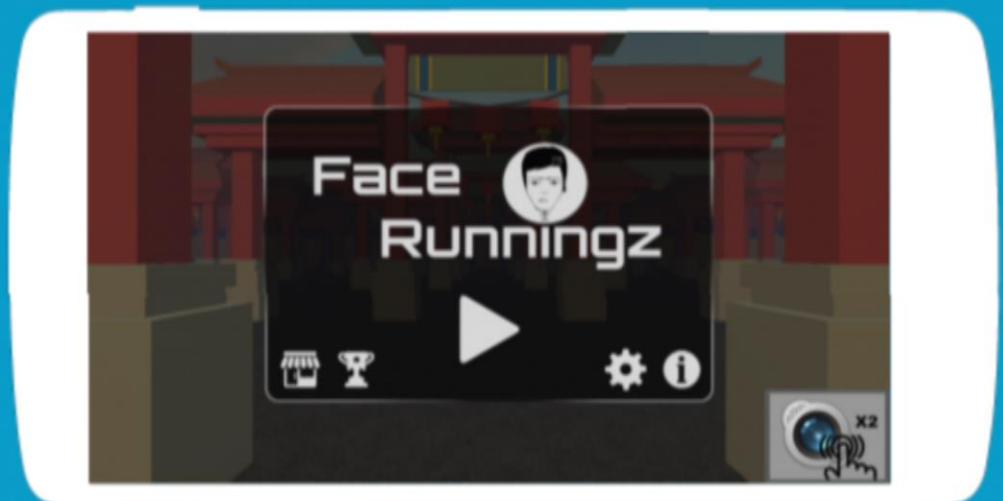
Exclude bibliography Off

Exclude matches Off

# Development of Interactivity Game using Facial Landmarks

FaceRunningz is motion-control game where it allows user to interact with an intermediate 3D graphic game environment using front camera to detect face movement without touching device within a distance while certain game control methods rely on face expression.

Available On:



## Why it needed?

- Lack of motion-control game
- Required specific hardware & costly

## Methodology

- Based on Agile development process
- Google Face API & Android Studio for facial landmarks detection
- Unity game engine for 3D game environment

## Features

### ■ Face Detection

Use facial landmarks detection and face expression recognizer as game controller

### ■ Item Store

Provide various of game customizations to unlock special abilities and extra game modes

### ■ High Score & Leaderboard

To manage and view high scores in numerous categories