HAND GESTURE CONTROL FIRST-PERSON SHOOTER GAME

GOH CHIANG CHENG

UNIVERSITI TEKNOLOGI MALAYSIA

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author’s full name : GOH CHIANG CHENG

Date of Birth : 6/1/1997

Title : HAND GESTURE CONTROL FIRST-PERSON SHOOTER GAME

Academic Session : 2020/2021

I declare that this thesis is classified as:

|  |  |  |
| --- | --- | --- |
|  | CONFIDENTIAL | (Contains confidential information under the Official Secret Act 1972)\* |
|  |  |  |
|  | RESTRICTED | (Contains restricted information as specified by the organization where research was done)\* |
|  |  |  |
| ✓ | OPEN ACCESS | I agree that my thesis to be published as online open access (full text) |

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Certified by: |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | SIGNATURE OF STUDENT |  | SIGNATURE OF SUPERVISOR |  |
|  |  |  |  |  |
|  | A17CS0048 |  | DR. JUMAIL BIN TALIBA |  |
|  | **MATRIX NUMBER** |  | **NAME OF SUPERVISOR** |  |
|  |  |  |  |  |
|  | Date: |  | Date: |  |

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my

opinion this thesis is suffcient in term of scope and quality for the

award of the degree of Bachelor of Computer Science (Graphics and Multimedia Software)”

|  |  |  |
| --- | --- | --- |
| Signature | : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Name of Supervisor | : | JUMAIL BIN TALIBA |
| Date | : |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

HAND GESTURE CONTROL FIRST-PERSON SHOOTER GAME

GOH CHIANG CHENG

A thesis submitted in fulfilment of the

requirements for the award of the degree of

Bachelor of Computer Science (Graphics and Multimedia Software)

School of Computing

Faculty of Engineering

Universiti Teknologi Malaysia

AUGUST 2021

DECLARATION

I declare that this thesis entitled *“Hand Gesture Control First-Person Shooter Game”* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

|  |  |  |
| --- | --- | --- |
| Signature | : | .................................................... |
| Name | : | GOH CHIANG CHENG |
| Date | : |  |
|  |  |  |
|  |  |  |

DEDICATION

This thesis is dedicated to all lecturers in Universiti Teknologi Malaysia (UTM) who did share their knowledge to me during my study in the university.

ACKNOWLEDGEMENT

I would like to express my appreciation to all the lecturers in School of Computing that did provides useful knowledge and guidance to me for completing my study in Universiti Teknologi Malaysia (UTM). I would also like to thank my supervisor, Dr. Jumail Bin Taliba for his encouragement, patience, and support. His guidance helped me a lot in deciding the title for this thesis and technique that is related to it.

In addition, I would like to express my gratitude to the PSM committee of School of Computing for providing the guideline and material that is needed for this thesis. The content provided by them are well-organized and easy to understand.

Finally, my thanks will be given to my friends and family for their continuous support and caring. Their support has motivated me to work harder on completing this thesis.

ABSTRACT

The purpose of this study is to develop a First-Person Shooter (FPS) game which can be control through hand gesture without the use of hardware such as Kinect sensor. The concept of the project is to implement the knowledge of image processing to detect the hand gesture that has been captured by the webcam. Hand gesture is a more natural communication medium that the player can send instruction to the computer in compare with the use of traditional input device such as mouse and keyboard. There will be two subparts during the development of the proposed system which is the FPS game development and hand gesture detection development. These two parts will be evaluated separately, and the final evaluation will be done again after the integration of the system. This project is expected to prove that the use of webcam as the hand gesture detection device does have the potential to decrease the cost that needed for the hand gesture sensor while provides the features which is similar like it.

ABSTRAK

Kajian ini dilakukan bertujuan menghasilkan satu permainan video berjenis penembak diri-pertama yang boleh dikawal melalui isyarat tangan tanpa menggunakan perkakasan seperti sensor kinect. Konsep projek ini adalah menggunakan pengetahuan pemprosesan imej untuk mendapatkan informasi isyarat tangan yang ditangkap oleh kamera web. Isyarat tangan adalah jenis medium komunikasi yang lebih mudah dan selesa yang boleh dilakukan oleh pemain untuk menberi arahan kepada komputer berbanding dengan medium komunikasi biasa seperti penggunaan tetikus dan papan kekunci. Sistem yang dicadangkan terdapat dua bahagian iaitu bahagian penghasilan permainan video berjenis penembak diri-pertama dan bahagian pengesan isyarat tangan. Kedua-dua bahagian akan dinilai secara berasingan dan penilaian yang akhir akan dibuat selepas bahagian tersebut digabungkan. Projek ini dijangka dapat membuktikan bahawa kegunaan kamera web sebagai pengesan isyarat tangan mempunyai potensi untuk mengurangkan kos untuk sensor isyarat tangan dengan memperoleh fungsi yang serupa.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
|  | TITLE | PAGE |

[DECLARATION ii](#_Toc69670080)

[DEDICATION iii](#_Toc69670081)

[ACKNOWLEDGEMENT iv](#_Toc69670082)

[ABSTRACT v](#_Toc69670083)

[ABSTRAK vi](#_Toc69670084)

[TABLE OF CONTENTS vii](#_Toc69670085)

[LIST OF TABLES ix](#_Toc69670086)

[LIST OF FIGURES x](#_Toc69670087)

[LIST OF ABBREVIATIONS xi](#_Toc69670088)

[LIST OF APPENDICES xii](#_Toc69670089)

[CHAPTER 1 INTRODUCTION 1](#_Toc69670090)

[1.1 Introduction 1](#_Toc69670091)

[1.2 Problem Background 2](#_Toc69670092)

[1.3 Project Aim 2](#_Toc69670093)

[1.4 Project Objectives 3](#_Toc69670094)

[1.5 Project Scope 3](#_Toc69670095)

[1.6 Project Importance 4](#_Toc69670096)

[1.7 Report Organization 4](#_Toc69670097)

[CHAPTER 2 LITERATURE REVIEW 7](#_Toc69670098)

[2.1 Introduction 7](#_Toc69670099)

[2.2 First-Person Shooter game 7](#_Toc69670100)

[2.2.1 Definition of FPS game 7](#_Toc69670101)

[2.2.2 Characteristic of FPS game 8](#_Toc69670102)

[2.2.3 Requirement of FPS game 8](#_Toc69670103)

[2.3 Hand Gesture 9](#_Toc69670104)

[2.3.1 Definition of Hand Gesture 10](#_Toc69670105)

[2.3.2 Hand gesture recognition 10](#_Toc69670106)

[2.3.3 Importance of hand gesture control in video game industry 11](#_Toc69670107)

[2.4 Existed System 11](#_Toc69670108)

[2.4.1 Webcam-based hand gesture control in Skyfall 12](#_Toc69670109)

[2.4.2 G-Shooting Gesture 13](#_Toc69670110)

[2.5 Comparison between existing systems 13](#_Toc69670111)

[2.6 Literature Review of Technology Used 14](#_Toc69670112)

[2.6.1 Open Source Computer Vision Library (OpenCV) 14](#_Toc69670113)

[2.6.2 Unity3D 15](#_Toc69670114)

[2.7 Chapter Summary 18](#_Toc69670115)

[CHAPTER 3 SYSTEM DEVELOPMENT METHODOLOGY 19](#_Toc69670116)

[3.1 Introduction 19](#_Toc69670117)

[3.2 Methodology Choice and Justification 19](#_Toc69670118)

[3.3 Phases of the Chosen Methodology 20](#_Toc69670119)

[3.4 Technology Used Description 22](#_Toc69670120)

[3.5 System Requirement Analysis 23](#_Toc69670121)

[3.6 Chapter Summary 24](#_Toc69670122)

[CHAPTER 4 SYSTEM DEVELOPMENT 27](#_Toc69670123)

[4.1 Introduction 27](#_Toc69670124)

[CHAPTER 5 SYSTEM TESTING AND DISCUSSION 37](#_Toc69670125)

[5.1 Introduction 37](#_Toc69670126)

[CHAPTER 6 CONCLUSION 38](#_Toc69670127)

[6.1 Review on the Final Output 38](#_Toc69670128)

[6.2 Potential Improvement 38](#_Toc69670129)

[REFERENCES 39](#_Toc69670130)

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TITLE | PAGE |

[Table 3‑1: Hardware requirement 24](#_Toc69620398)

[Table 3‑2 Software requirement 24](#_Toc69620399)

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE |

[Figure 2.1: Maze War (Grimshaw et al., 2011) 9](#_Toc73288068)

[Figure 2.2: The framework of hand gesture detection (Ma’asum, Sulaiman, Saparon, 2015) 11](#_Toc73288069)

[Figure 2.3: Hand gesture control in Skyfall 12](#_Toc73288070)

[Figure 2.4: Sample gameplay of G-shooting Gesture 13](#_Toc73288071)

[Figure 2.5: How computer understand about the image (Kaehler and Bradski, 2016) 15](#_Toc73288072)

[Figure 2.6: Sample interface of Unity3D 16](#_Toc73288073)

[Figure 2.7: Unity Asset Store 17](#_Toc73288074)

[Figure 2.8: Multiplatform available in Unity3D 17](#_Toc73288075)

[Figure 3.1: Project methodology 20](#_Toc73288076)

[Figure 3.2: Phases of an iteration in agile methodology 21](#_Toc73288077)

[Figure 3.3: Concept image of the technology use in the project 22](#_Toc73288078)

[Figure 3.4: Flow of hand gesture detection and recognition in OpenCV 23](#_Toc73288079)

[Figure 4.1: Components in Unity3D 29](#_Toc73288080)

[Figure 4.2: Scene Design for Main Menu and FPS Game in Unity3D 30](#_Toc73288081)

[Figure 4.3: Login Interface 31](#_Toc73288082)

[Figure 4.4: Main Menu Interface 32](#_Toc73288083)

[Figure 4.5: Choose Stage Interface 33](#_Toc73288084)

[Figure 4.6: Leaderboard Interface 34](#_Toc73288085)

[Figure 4.7: FPS Game Interface 35](#_Toc73288086)

LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| FPS | - | First-Person Shooter |
| PC | - | Personal Computer |
| NPC | - | Non-Player Character |
| 3D | - | Three Dimensions |
| API | - | Application Programming Interface |
| OpenCV | - | Open Source Computer Vision Library |
| IDE | - | Integrated Development Environment |
| OS | - | Operating System |
| RAM | - | Random Access Memory |
| UI | - | User Interface |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

LIST OF APPENDICES

|  |  |  |
| --- | --- | --- |
| APPENDIX | TITLE | PAGE |

[Appendix A 41](#_Toc69591017)

# INTRODUCTION

## Introduction

According to Kuikkaniemi et al. (2010, April), first-person shooter (FPS) games is a well-known video game category which focus on the first-person perspective and provides a gaming platform of firearm-based combat. The first-person perspective is set on an avatar’s point of view which allows the player to experience the game through the view of the avatar to provide a better immersive feel that the player can observe and interact with the environment around the avatar (Denisova and Cairns, 2015, April). Normally, the FPS games will let the player able to see an arm (or pair of arms) which is either empty-handed or holding any type of weapon such as pistol or knife (Grimshaw, Charlton and Jagger, 2011). This weapon will let the player able to interact with the target in the games which can be either programmed character or player. One of the simplest interactions is to aim and shoot the target until the target is taken down.

Hand gestures is a simple communication method for the user to interact with the computer and no advance training is needed for the user to understand the way to communicate with the computer (Hamza, Anand, Shvhare and Gaurav, 2017). By using hand gestures interaction, the user will only need to know the pre-defined hand shape and movement that match the instruction set up by the programmer to execute certain command to the computer. According to Dardas, Silva and El Saddik (2014), The advantages of hand gestures control in video game is that the user can interact with the computer without physical contact and make the interaction process become more natural, immersive and comfortable.

## Problem Background

The improvement of technology in computer science makes the customer in gaming industry wants to play the computer games that involved body movements which is close to the actual ways the people will do in real life (Roccetti, Marfia and Semeraro, 2012). Some user feels bored in playing games through traditional interface such as keyboard and mouse. Hand gesture control can let the user play the video game with some predefined hand pattern and movement which is more realistic and entertaining to play. This is because hand gestures are considered as the most natural and flexible communication media to transfer the command to the computerized devices in compare with the other interface such as keyboard or mouse (Chen, Lin and Yang, 2010, June). According to Roccetti et al. (2012), although the price of the equipment that act as the sensor of the hand gestures such as Leap Motion Controller and XBOX360 Kinect Sensor has gradually decreased, it is still more worthy if there is a solution to make the simple webcam act as the device to capture the hand gesture input. Although webcam cannot provide the depth information of the environment, but the use of image processing can still detect the hand gesture with the information provided by the webcam in real-time manner (Hamza et al., 2017).

## Project Aim

The aim of this project is to develop a first-person shooter game which can be played through hand gesture with the input from webcam.

## Project Objectives

The objectives of the project are:

1. To analyse the requirements of first-person shooter games controlled using hand gestures.
2. To design a personal computer (PC) game controlled by hand gestures.
3. To implement the image processing algorithm through input from webcam.
4. To evaluate the potential of webcam that can replace the use of other hand gesture sensor.

## Project Scope

The scopes of the project are:

1. The game produced will be in single player and personal computer based which will fully controlled by hand gesture throughout the whole gaming process.
2. The hand gestures input will only focus on hand shape and movement only.
3. The character in the game would not be able to move and they need take down all the enemies in the scene to go to the next stage. A shield command will be provided to block enemies attack.
4. The game will in stage form which let the player can start at any stage that they clear before.
5. The image input will be gathered only through webcam without any others image input devices.

## Project Importance

The project is important to use image processing algorithm to identify the hand gesture that has been captured by the webcam to interact with the FPS game. The hand gesture should be more relevant that can let the player feel comfortable and enjoy the gaming process. The main idea of the project is to use webcam approached to offer the functionality of Kinect sensor in order to decrease the cost needed for game which is controlled using hand gesture.

## Report Organization

Chapter 1 is the introduction of the project and it has been described in the section above. The next chapter is Chapter 2 which describes the literature review. Chapter 3 will explain about the project methodology and the proposed system will be explain in Chapter 4. Chapter 5 will make a conclusion about the project.

Chapter 2 is the literature review of the project. This chapter describes about first-person shooter (FPS) game, hand gestures and the comparison with existed system. The technology used will be review at this chapter.

Chapter 3 explain about the project development methodology. This is the project guideline that use to define the procedure on implementing and developing the project to achieve the aim of the project.

Chapter 4 will state the requirement analysis and the design of the project. The design will be described in this section includes project design and interface design.

Chapter 5 concludes the project by describing the expected achievements of the project objectives, constraints and state the planning for PSM 2.

# LITERATURE REVIEW

## Introduction

This chapter describes the terms of First-Person Shooter (FPS) game and explain about hand gesture which includes the importance of hand gesture in video game. Lastly, some previous works which is similar like the project will be reviewed and compare with the proposed system.

## First-Person Shooter game

The subsections will describe the definition, characteristic and requirement of FPS game.

### Definition of FPS game

According to Hitchens (2011), FPS is a genre of video game which use first-person viewpoint as scene display that is heavily focus on the combat between player and player or player and non-player character (NPC) which is mainly using firearms. The use of ranged weapon to start the combat between the player and the opponent will create a more rapidly changing environment which required the player to react fast and accurately so that they can take down the opponent as fast as possible to prevent taking too much damage from the opponent (Colzato, Van Leeuwen, Van Den Wildenberg, Hommel, 2010). FPS game does provide a platform for those people who likes firearm to be able to use it in a virtual environment which fulfil their hype on using it.

### Characteristic of FPS game

FPS games are normally considered as a violent video game due to the involvement of guns and the action of taking down enemy through firearms (Weber et al., 2009). The first main genre of FPS games is single player mode which provides a storyline for the player to explore and immerse themselves into the experiences of the avatar they are controlling. According to Cardamone, Yannakakis, Togelius and Lanzi (2011, April), this kind of FPS game will have a limited playing time around 10 hours depends on the budget and the storyline the developer wishes to present to the player. The player will only face NPC which has applied the computational intelligence techniques as the basic of its behaviour. The second main genre is multiplayer mode, the player needs to compete against each other to take down the human controlled avatar that are not same team as them or fighting against the NPC together to achieve the mission that is being given by the system (Zhou, Miller and Bassilious, 2008, May).

### Requirement of FPS game

According to Therrien (2015), the FPS game is always bind to the terms, “perspective”, “3D” and “simulation”. The mainframe game Maze War (1974) which is considered as the earliest FPS genre game is a good example that explain how a game can be defined as the FPS game. In term of perspective and simulation, the player should have the ability to perform task with the viewpoint of the avatar inside the FPS game depends on the command given to the avatar as in Maze War. For instance, the scene of Maze War will change when the moving instruction has been applied or the player give the command to fire a shot, a bullet will also be fired. The ”3-D” term is the visual representation form of the environment of the FPS game. The scene should be able to provides some interaction that is close to the understanding of player on the real 3D world such as the wall will block the movement and view of sight of the player as in Maze War (Grimshaw et al., 2011) which has been shown in Figure 2.1.

A picture containing computer, light

Description automatically generated

Figure .: Maze War (Grimshaw et al., 2011)

Besides the fundamental requirement of FPS game, the game content which includes maps and levels into the FPS game is crucial to optimize the player experience contemporary (Cardamone et al., 2011, April). The FPS game that the player only can go through the same map and shoot the same types of bot will not be able to provides the joy that the player needed from the game. Zhou et al. (2008, May) did state that the internet delay must be considered for the multiplayer FPS game since this will affect heavily on the result of the game.

## Hand Gesture

The subsections will describe the definition of hand gesture, hand gesture recognition and the importance of hand gesture in video game industry.

### Definition of Hand Gesture

According to Chen et al*.* (2014), gesture is a result of a human behavior and a way that human use to express their emotion and feeling. There is two main type of gesture which is hand gesture and body gesture that both can be use on sending useful message to others. Hand gesture is a type of compressed information which can only be understood by those who knows how to decode the gesture presented (Ma’asum, Sulaiman, Saparon, 2015). The change in the shape and movement of the finger will define the hand gesture but the same hand gesture may have different meaning in terms of the understanding and knowledge of the sender and the condition of the situation.

### Hand gesture recognition

There is three main process that will gone through to detect and recognize the hand gesture presented by the user which are camera module, detection module and feature extraction module (Ma’asum et al., 2015) while the flow of the steps has shown in Figure 2.2. In camera module, the image contains the hand gesture will be capture by the webcam and converted into digital information which will be use on detection module to retrieve the data of the threshold images. The system will identify the region in the image that needed to be focus on which is the hand region so that the data in that specific region will be emphasized and take as the primary data that will be use during the Feature Extraction Module (Chen et al., 2014). During the Feature Extraction Module, the feature of the hand region will be extracted and use as the description of the hand gesture. After these three modules, the feature information will be given to the program to compare with the data inside of the feature database to calculate the similarity between each dataset and identify which hand gesture has been used in the image.

A screenshot of a cell phone

Description automatically generated

Figure .: The framework of hand gesture detection (Ma’asum, Sulaiman, Saparon, 2015)

### Importance of hand gesture control in video game industry

The use of gesture-based interfaces which is closer to the real-life interaction has become one of the primary factors that can determine the success of a video games (Roccetti et al., 2012). The user starts to find the user interface that is less reliable on traditional interfaces like joystick, keyboard or mouse but wants to play the video games with the ways that is far more natural and closer to the action in the reality. For instance, they wish to grab something in the video game with grabbing hand but not left click the mouse or looks around by moving their head but not moving the mouse. According to Chen et al. (2010, June) although a glove-based approach did propose in some video games, but the user still not satisfy with the needs on wearing the glove since this input interface required the player to have addition equipment that they need to wear on.

## Existed System

In this section, 2 existed system which is webcam-based hand gesture control in Skyfall and G-shooting Gesture will be introduced.

### Webcam-based hand gesture control in Skyfall

Skyfall is a kind of web-based game that the player can play it using the browser. The ways to play the game is to control the paddle left and right to collide with the ball that is falling down. White ball and green ball will give the player 10 points while red ball will decrease 10 points from the player. The standard Skyfall interface is using mouse to control the movement of the paddle, but the developer did add in the hand gesture control that get the image input through webcam to detect the hand movement of the player to control the paddle. The sample image has shown in Figure 2.3.

This application use TensorFlow object detection application programming interface (API) as the machine learning library and trained with the models of the hand that the application needs to detect and recognize. The detected hand gesture will be sent by the API to the game interface as the input to control the paddle without the needs on using mouse.

A picture containing computer, sitting, computer, star

Description automatically generated

Figure .: Hand gesture control in Skyfall

### G-Shooting Gesture

G-Shooting Gesture is a mobile game which is able to be download in Google Play Store. It is a kind of FPS game that the player can control the gun in the game through the camera of their mobile phone by using hand gesture. The hand gesture needed for the controlling is using their right hand’s index finger. The cursor in the application will follow the movement of the index finger and start shooting when the index finger has stop moving. Figure 2.4 is the image which show the sample gameplay of G-shooting Gesture.

A picture containing person, indoor, holding, person

Description automatically generated

Figure .: Sample gameplay of G-shooting Gesture

## Comparison between existing systems

The 2 existing system that stated in Chapter 2.4 does have the similar concept with the proposed system in this project which is detect the hand gesture of the user without the assistance from motion sensor hardware. The image of the hand gesture will capture through normal camera and recognized by the system through image recognition. The differences between the two existing system and the proposed system are the platform of the game, the proposed system will be a personal computer (PC) game.

## Literature Review of Technology Used

This section will define the technology and platform that will be uses to develop the proposed system which is OpenCV library and Unity3D.

### Open Source Computer Vision Library (OpenCV)

OpenCV is an open source library mainly written in C and C++ which has been further developed into integrate with others programming interface such as Python, Java and Matlab. According to Kaehler and Bradski (2016), the purpose of OpenCV is to enhance the usage of computer vision and artificial intelligence which can be processed in real-time application. OpenCV is used to fulfill the aim of computer vision by understanding the data contains in an image and convert it into the knowledge on making some sort of decision. Figure 2.5 is an example of the needs of computer vision since the computer will not know that the image is a car mirror but just a matrix contains values. OpenCV will be able to determine that the image input is a car mirror through image processing or even machine learning.

A picture containing toy

Description automatically generated

Figure .: How computer understand about the image (Kaehler and Bradski, 2016)

### Unity3D

Unity3D is a game engine that is being developed by David Helgason, Joachim Ante and Nicholas Francis which provides an integrated development environment (IDE) for developing video game (Haas, 2014). Unity3D is an open source IDE which provides the opportunities for independent developer to be able to produce interaction media without the needs on spending money on buying the license. Figure 2.6 is the sample interface of Unity3D and there are 2 core components that makes Unity3D a successful game engine which is the assets pipeline and multiplatform compatibility.

A screenshot of a computer

Description automatically generated

Figure .: Sample interface of Unity3D

The asset pipeline in Unity3D provides the possibility for the developer on taking the assets that has been created by others developer that shared on the Unity Asset Store to integrate with their project. These assets can be 3D model or script that the owner choose to publish in the Unity Asset Store as shown in Figure 2.7 with price or sometimes it will be free. This provides a platform for the developer to share their product, learn from others or even decrease the development of their video game since they may not need to produce those 3D model or script themselves and focus more on the other content of the games. There is also some asset is use for the tutorial so that the new developer can be familiar with Unity3D.

A screenshot of a cell phone

Description automatically generated

Figure .: Unity Asset Store

Haas (2014) did states that Unity has become the favourite of the game developer is because Unity3D did provides the functionality on publishing the video game in different platform by using the same code and asset. These platforms included mostly all the current popular platform such as iOS, Android, Windows, Mac OS X and Linux. Figure 2.8 has displayed the multiplatform available in Unity3D which can decrease the needs of reprogramming the project so that the application can be run in different platform.

A screenshot of a cell phone

Description automatically generated

Figure .: Multiplatform available in Unity3D

## Chapter Summary

Subchapter 2.2 had provided an explanation about the terms FPS game and what is the characteristic and requirement of an FPS game. FPS game is a genre of games that has many fans and is well-known among the gamers.

Subchapter 2.3 discuss about the definition of hand gesture and the ways to recognize hand gesture. The importance of hand gesture control in video game will also state in this subsection. 3 main components in recognizing hand gesture are camera module, detection module and feature extraction module. Hand gesture control is important in video game because of its real-life interaction which provides a more natural ways to interact with the video games.

Subchapter 2.4 list two existed system which is similar to the proposed project. One of the existed systems is webcam-based hand gesture control in Skyfall and the others is G-Shooting Gesture.

In Subchapter 2.5, the comparison between the existed system and proposed system has been done. The similarity of the system is that both using webcam as the input of the hand gesture, but the differences is the platform of the video games.

Subchapter 2.6 is the literature review of the technology used which is OpenCV and Unity3D. OpenCV will use on recognizing the hand gesture while Unity3D is the platform that use to create the FPS game.

# SYSTEM DEVELOPMENT METHODOLOGY

## Introduction

This chapter will describe the system development methodology that has been used on developing the system which is agile development methodology.

## Methodology Choice and Justification

The methodology used in this project is agile development methodology. The agile development methodology has decreased the potential risk to the project which includes change in requirement and adding new functionality. Agile method has successfully improved the quality of the system due to the small iteration of testing and requirement checking during system development. The FPS game and hand gesture detection has been tested a lot of times and many new functions has been added to enhance the performance of the system. Figure 3.1 shows the flow of the Project Methodology.

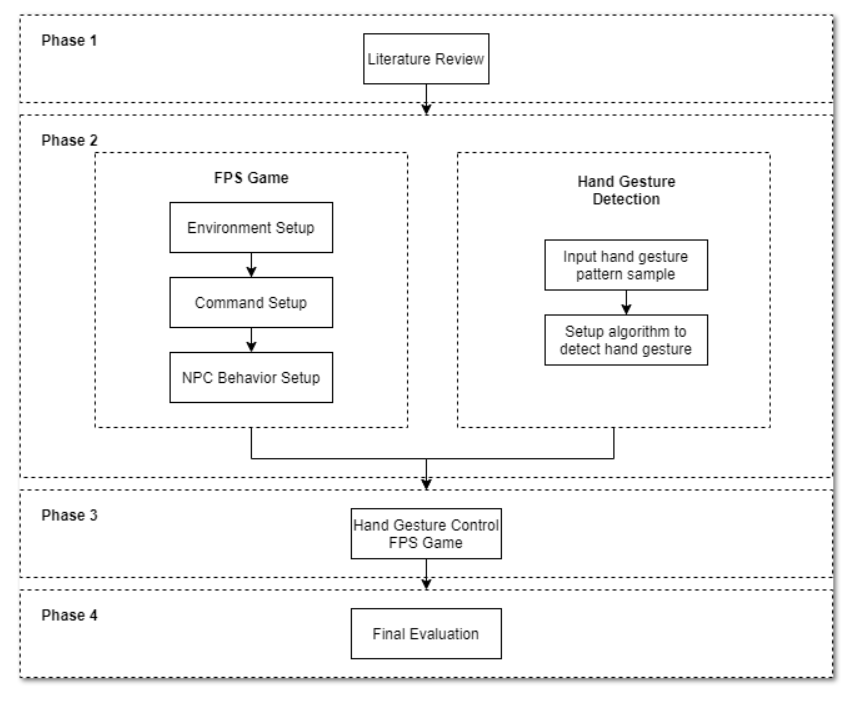


Figure .: Project methodology

## Phases of the Chosen Methodology

The 5 main phases in each iteration of the agile methodology are determine requirement, system design, implementation, testing and make improvement that has been shown in Figure 3.2. These 5 phases have been gone through in each iteration and the iteration does not change if the result does not fulfil the requirement.

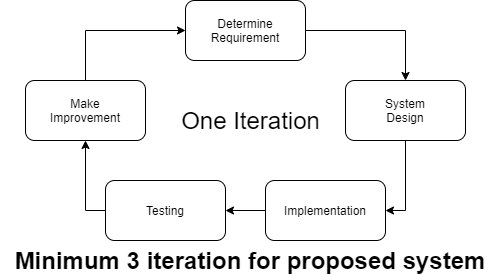


Figure .: Phases of an iteration in agile methodology

The first phase is determining the requirement of the system which means that the functionality and the expected performance that the current iteration of system needs to fulfil have been identify. The system has been designed according to the requirement that did define at previous phase. Due to the use of agile methodology, the requirement at this spot has become the minor achievement of current iteration and increased after the next iteration. The system has been developed at implementation phase through coding or user interface creation which involve the actual use of computer to implement the algorithm.

The user has tested the iteration of the system to determine the performance of current system. The system has completed the improvement before going to the next iteration with the same cycles according to the user feedback. The proposed system does have two subpart which is the hand gesture detection section and FPS game section. Each section has gone through the iteration stated above and the combination of the subsection were going through again the same iteration.

## Technology Used Description

There are two main components in this project which is OpenCV and Unity3D. Figure 3.3 is the concept image of the technology which has been used in this project. OpenCV is the open vision library which use to detect and recognize the hand gesture that will be input from the webcam. The flow of the process is shown on Figure 3.4. The first step is capturing the image through the webcam and then the skin color detection has been done by using OpenCV. The skin color detection will identify the human body parts and further process will be done to remove the human body parts which is not the hand. Later on, the hand gesture will be filter into a grayscale image to make the hand gesture recognition system can be done more accurately and efficiently.

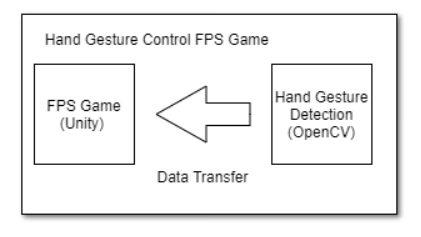


Figure .: Concept image of the technology use in the project

A screenshot of a cell phone

Description automatically generated

Figure .: Flow of hand gesture detection and recognition in OpenCV

For the creation of FPS using Unity3D, a lot of assets have been created and insert into the scene to make the stage of the FPS game. The Unity3D does compile the application using its project build function. Microsoft Visual Studio has been used as the platform to modify the C# script in Unity3D. The input was gathered from OpenCV and the avatar in the game did act according to the input retrieve from OpenCV.

## System Requirement Analysis

There is two main requirement that have been analyzed which is the hardware requirement and software requirement. Unity3D is the main parts of the development of FPS game so the Operating System (OS) used did support Unity3D with version 2019.1. Table 3.1 list out the hardware requirement while Table 3.2 is the software requirement for the project.

Table 3‑1: Hardware requirement

|  |  |
| --- | --- |
| Component | Specification |
| OS | Windows 7 or above  macOS 10.12 or above |
| RAM | 4 GB DDR3 RAM |
| Graphic Card | Graphics card with DX10 (shader model 4.00) capabilities |

Table 3‑2 Software requirement

|  |  |
| --- | --- |
| Software | Specification |
| Game Engine | Unity 2019.1 |
| C# Script Editor | Microsoft Visual Studio 2017 |

## Chapter Summary

Subchapter 3.2 has state the methodology choice for this project. Agile methodology has been selected for this project because the FPS game performance and the hand gesture detection can be evaluated separately.

Subchapter 3.3 show the phases for the chosen methodology. There are 5 phases for each iteration which is determine requirement, system design, implementation, testing and make improvement. The project will only go to the next iteration after the expectation and requirement for current iteration has been fulfilled.

Subchapter 3.4 describe the technology used in this project which is OpenCV and Unity3D. OpenCV is use on hand gesture detection while Unity3D is use on constructing the whole FPS game.

Subchapter 3.5 state the system requirement for the development of the project. Hardware requirement is determined by the compatibility with the Unity3D and OpenCV version.

# SYSTEM DEVELOPMENT

## Introduction

This chapter will provide the details about the function and addon used during the development of the project. The system development will be separated into 2 parts, which is the FPS Game part and Image Processing.

## Requirement Analysis

The functional requirement and non-functional requirement that the completed system has been fulfilled has been listed at below.

### Functional Requirement

1. The completed system is able to detect hand gesture through webcam.
2. The completed system is able to provides an FPS genre game for the player to play.
3. The avatar in the FPS game does react correctly to the predefined hand gesture of the player.
4. The player is able to shoot down NPC through hand gesture.
5. The completed system does provide a tutorial for the player.
6. The completed system does pinpoint the NPC that the player is currently aiming.

### Non-Functional Requirement

1. The completed system is able to recognize more than 3 types of hand gesture.
2. The completed system is able to response to user’s hand gesture in less than 2 second.
3. The completed system does provide 3 stages for the player.
4. The completed system is able to provide leaderboard for the user to track their result.

## Design

The whole system has been constructed under the game engine, Unity3D. All of the contents in the FPS game will be build and design using Assets in Unity and the behavior will be programmed using C# programming language. The image processing features will be implemented in Unity3D using a well setup wrapper, OpenCV+ Unity while the login features and leaderboard will be provided with the use of server platform, Microsoft Azure Playfab. The relationship of these components has been shown on Figure 4.1.

Graphical user interface

Description automatically generated

Figure .: Components in Unity3D

There will be 4 scenes inside Unity3D which is Main Menu, Easy Stage, Normal Stage and Hard Stage. Since Easy, Normal and Hard Stage scene has the identical interaction setup, so in Figure 4.2, only one graph will use as the reference of all 3 Stage scene.

In Main Menu scene, there is 2 Game Object which is Main Menu Manager and Playfab Manager that will use to control the interface display but getting the command of the player. Main Menu Manager will swap between the 4-user interfaces (UI) that starts with Login UI, Main Menu UI, Choose Stage UI and Leaderboard UI. The Playfab Manager will use to retrieve or send the data related to player information and leaderboard value from the Play Fab online server. The Main Menu needs these data to check user login, register user and display leaderboard.

For the FPS Game Scene, the Game Manager will constantly be checking the condition of the Player in order to display the correct user interface like Game Over UI, Game Pause UI and Game Complete UI. The Game Manager also helps to determine the timing for the Player to move to the next area. The Image Detection script will send the result of the command that has been determined by analyzing the pattern of the hand gesture using OpenCV+Unity functions to the Player in order to do the action like firing the shot or changing target. Both Player and Enemy will be able to deal damage to each other’s.

Diagram

Description automatically generated

Figure .: Scene Design for Main Menu and FPS Game in Unity3D

## Interface Design

The explanation of the interface design will be separated into 2 sub section which is Main Menu and FPS Game. Main Menu section contains all the selection interface for the player to choose their desired interface. In Main Menu, the player can login, register, choose the stage of FPS game and view leaderboard. The FPS Game section will describe about the element of the user interface when the FPS game is ongoing.

### Main Menu

There are 4 interfaces in the Main Menu section which is Login Interface, Main Menu Interface, Choose Stage Interface and Leaderboard Interface.

#### Login Interface

In the Login Interface, the player is able to login their account by inputting the username and password. New player is also able to register their account by filling in their username and password in the register field. The user will be able to instantly login and go to the Main Menu Interface if their register successfully. The username is the primary key for each user, therefore any username input that has existed in the Playfab player database will be rejected during the registration. Figure 4.3 shows the Login Interface.

A picture containing text, monitor, screenshot

Description automatically generated

Figure .: Login Interface

#### Main Menu Interface

There will be 5 selection buttons for the Main Menu Interface, which is Start Game, Choose Stage, Tutorial, Leaderboard and Back as shown in Figure 4.4. The player will be able to start the FPS game in Easy stage when pressing the Start Game button or they can choose the difficulty of the stage by pressing the Choose Stage button. The Tutorial button will bring the player to Tutorial Stage. The player can view the leaderboard of the game for all 3 stages by pressing the Leaderboard button. Back button will transfer the player back to the Login Interface.

Graphical user interface

Description automatically generated

Figure .: Main Menu Interface

#### Choose Stage Interface

The Choose Stage Interface provides all 3 difficulties of stage for the player, which is Easy, Normal and Hard. Each button will lead the player to the respective stage. The player also able to go back to the Main Menu Interface by pressing the Back button. Figure 4.5 shows the design of Choose Stage Interface.

A picture containing text, monitor, light

Description automatically generated

Figure .: Choose Stage Interface

#### Leaderboard Interface

Figure 4.6 shows the design of the Leaderboard Interface. There will be 3 interfaces with similar design for all the difficulties. The player is able to swap between all 3 leaderboards interface by pressing the arrow button. The value of the Leaderboard will be retrieved from the Playfab online leaderboard server.



Figure .: Leaderboard Interface

### FPS Game

Figure 4.7 shows the FPS Game Interface with labels. Label 1 is the text elements which inform the player about the difficulty of current stage. Label 2 is an object which represent the enemy. The player needs to take down all of the enemy in the scene to proceed to next area. Label 3 is the countdown value of the next attack timing of the enemy. When the value goes to 0, the enemy will fire an electric shock towards the player. Label 4 is the text elements which shows the total time the FPS game has go through. The shorter the time to complete the FPS game, the highest the ranking the player will be on the leaderboard. Label 5 is the video that has been capture from the webcam of the player. Label 6 is the remaining health points of the player. When the health points go to 0, the FPS game will be over.

A screenshot of a video game

Description automatically generated

Figure .: FPS Game Interface

## Chapter Summary

Subchapter 4.2 listed down those functional requirement and non-functional requirement that has been fulfilled by the completed system.

Subchapter 4.3 describe about the components that used to construct the completed system. These components are Unity3D, OpenCV+ Unity and Microsoft Azure Playfab. The ways these components implemented has been further explained in this section.

Subchapter 4.4 shows and explain about the interface design of the completed system. There are 2 main interfaces in the system which is Main Menu and FPS Game.

# RESULT, TESTING AND DISCUSSION

## Introduction

This chapter will discuss about the coding of the system and the results get from the program. The testing result will also be described in this chapter.

## Coding of system’s main function

The coding that uses in the whole system is C# programming language. Unity provides the feature that the C# script can be embedded to certain GameObject in the scene in order to define the behavior of the system at certain scene. The 4 primary C# script setups in this project is “Main Menu Manager”, “PlayFab Manager”, “Game Default Setup Manager” and “FPS Game Manager”. Figure 5.1 shows the main C# script for the whole system.

A picture containing table

Description automatically generated

Figure .: Main C# Script int the System

### Main Menu Manager

This C# scripts determine the user interface that will be display according to the user selection. The first interface the user will be go through is “Login User Interface” which required the user to either login using existed username and password or register a new account for the system. “Main Menu Manager” contains all the function which will be used to switch between the user interface depending on the button that has been pressed by the user. The specific function in “Main Menu Manager” will be called by the OnClick() function provided in each button. Figure 5.2 shows the coding of “Main Menu Manager” while Figure 5.3 is the public variable which will be display by the inspector panel in Unity.

Graphical user interface, application, Word

Description automatically generated

Figure .: Coding of "Main Menu Manager" script

Graphical user interface, text, application, email

Description automatically generated

Figure .: Inspector panel of "Main Menu Manager" script in Unity

### PlayFab Manager

The “PlayFab Manager” script is used to setup the connection between PlayFab server and the system. This connection provides the function for the system to get and send user data which is the username and password to the PlayFab server in order to authenticate the user during the login or register process. This script also needed for the “Leaderboard User Interface” since every record for the FPS game will also save in the PlayFab server. The GameObject containing this script will be called in Unity using DontDestroyOnload() function so that it can still be able to exist after the scene have changed from “Main Menu” scene to the “Stage” scene. Figure 5.4 shows the coding of “Playfab Manager” while Figure 5.5 is the inspector panel of “Playfab Manager” script in Unity.

Graphical user interface, text, application, Word, email

Description automatically generated

Figure .: Coding of "Playfab Manager" script

Graphical user interface

Description automatically generated

Figure .: Inspector panel of "Playfab Manager" script in Unity

### Game Default Setup Manager

The “Game Default Setup Manager” save the variable which will be used throughout the system which is the “thresh” value. the “thresh” value is the constant that will be use by the image detection script during the FPS game and it can be change by the user in “Setting User Interface”. By calling the DontDestroyOnload() function on the GameObject that this script attached on, the “thresh” value will maintain the value that the user has changed to ensure the image detection process works fine for the user’s device. Figure 5.6 is the C# coding in “Game Default Setup Manager” script while Figure 5.7 is the interface of “Game Default Setup Manager” in the inspector panel of Unity.

Graphical user interface, text, application, email

Description automatically generated

Figure .: Coding of "Game Default Setup Manager" script

Graphical user interface, application

Description automatically generated

Figure .: Inspector panel of "Game Default Setup Manager" script in Unity

### FPS Game Manager

“FPS Game Manager” is the C# script used to determine the current progress of the FPS game. This script will continue tracing the condition of the game which includes the time used, player health point and the drone amount currently existed in the scene. “FPS Game Manager” script will transfer the player in the game to the next area when the drone in the previous area has confirmed to be terminated. When the player decides to pause the game, “FPS Game Manager” script will determine this instruction and temporary stop the time count throughout the gaming process which eventually freeze the movement and behavior of the drone in the FPS game. Figure 5.8 shows the coding in “FPS Game Manager” script and Figure 5.9 shows the interface value of “FPS Game Manager” script inside inspector panel in Unity.

Graphical user interface, text

Description automatically generated

Figure .: Coding of "FPS Game Manager" script

Graphical user interface, table

Description automatically generated

Figure .: Inspector panel of "FPS Game Manager" script in Unity

## System results

## Testing

## Chapter Summary

# CONCLUSION

## Introduction

This section will determine the output of the whole project and state the feature that has been successfully implemented.

## Achievement

This section will discuss about the improvement that can be done on the project to make it become a more complete system which can even have the potential to be commercialize.

## Future improvement

REFERENCES

Cardamone, L., Yannakakis, G. N., Togelius, J., & Lanzi, P. L. (2011, April). Evolving interesting maps for a first person shooter. In *European Conference on the Applications of Evolutionary Computation* (pp. 63-72). Springer, Berlin, Heidelberg.

Chen, W. H., Lin, Y. H., & Yang, S. J. (2010, June). A generic framework for the design of visual-based gesture control interface. In 2010 5th IEEE Conference on Industrial Electronics and Applications (pp. 1522-1525). IEEE.

Chen, Z. H., Kim, J. T., Liang, J., Zhang, J., & Yuan, Y. B. (2014). Real-time hand gesture recognition using finger segmentation. *The Scientific World Journal*, *2014*.

Colzato, L. S., Van Leeuwen, P. J., Van Den Wildenberg, W., & Hommel, B. (2010). DOOM'd to switch: superior cognitive flexibility in players of first person shooter games. Frontiers in psychology, 1, 8.

Dardas, N. H., Silva, J. M., & El Saddik, A. (2014). Target-shooting exergame with a hand gesture control. Multimedia tools and applications, 70(3), 2211-2233.

Denisova, A., & Cairns, P. (2015, April). First person vs. third person perspective in digital games: do player preferences affect immersion?. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 145-148).

Grimshaw, M., Charlton, J., & Jagger, R. (2011). First-person shooters: Immersion and attention.Eludamos. Journal for Computer Game Culture, 5(1), 29-44.

Haas, J. K. (2014). A history of the unity game engine.

Hamza, A., Anand, R., Shivhare, P., & Gaurav, A. (2017). Hand Gesture Recognition Applications. International Journal of Interdisciplinary Research, 13, 2073-2075.

Hitchens, M. (2011). A Survey of First-person Shooters and their Avatars. Game Studies, 11(3), 96-120.

Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.".

Kuikkaniemi, K., Laitinen, T., Turpeinen, M., Saari, T., Kosunen, I., & Ravaja, N. (2010, April). The influence of implicit and explicit biofeedback in first-person shooter games. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 859-868).

Ma'asum, F. F. M., Sulaiman, S., & Saparon, A. (2015). An overview of hand gestures recognition system techniques. In *IOP Conference Series: Materials Science and Engineering* (Vol. 99, No. 1, p. 012012). IOP Publishing.

Roccetti, M., Marfia, G., & Semeraro, A. (2012). Playing into the wild: A gesture-based interface for gaming in public spaces. Journal of Visual Communication and Image Representation, 23(3), 426-440.

Weber, R., Behr, K. M., Tamborini, R., Ritterfeld, U., & Mathiak, K. (2009). What do we really know about first-person-shooter games? An event-related, high-resolution content analysis. *Journal of Computer-Mediated Communication*, *14*(4), 1016-1037.

Therrien, C. (2015). Inspecting video game historiography through critical lens: Etymology of the first-person shooter genre. *Game Studies*, *15*(2).

Zhou, Q., Miller, C. J., & Bassilious, V. (2008, May). First person shooter multiplayer game traffic analysis. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (pp. 195-200). IEEE.

Appendix A