VOCAB DEFENDER : A TYPING SHOOTING GAME

FARHAN AKMAL BIN YAHYA

VOCAB DEFENDER : A TYPING SHOOTING GAME

FARHAN AKMAL BIN YAHYA

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Software Development)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

[Vocab Defender : A Typing Shooting Game]

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : _____ Date : 21/6/2021

(FARHAN AKMAL BIN YAHYA)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : _____ Date : 21/6/2021

(TS DR IBRAHIM BIN AHMAD)

DEDICATION

Praise to Allah SWT The Almighty

To my beloved parents, thank you for all your great support and understanding. Thank you for all encouragement, motivation and always be with me.

To my supervisor, Ts. Dr. Ibrahim Bin Ahmad, thank you for supporting me and guiding me go through the project development process and this report writing.

To my classmates and friends who assist me, thank you for providing feedback, suggestions, and criticism on my project in order for me to improve as a developer.

Finally, I'd like to thank Universiti Teknikal Malaysia Melaka for providing me with the chance and platform to improve my skills and knowledge in order to complete this project.

ACKNOWLEDGEMENTS

Grateful to Almighty Allah S.W.T and His bless for giving me strength and ability to complete the project in time provided with great satisfaction

I would like to thank my supervisor, Ts. Dr. Ibrahim Bin Ahmad. My sincerely gratitude to him for guiding me through the development process of this project from report writing to the game development phase.

I am also thankful from the core of my heart to my beloved parents and family who always support and love me.

Thanks for all concern and assistance.

ABSTRACT

This project were focusing on the development of Vocab Defender : The Typing Shooting Game which is aiming to improve player's typing skill and expose player with ICT terms used in daily life. Information and communication technologies (ICTs) are increasingly being used in educational settings to offer students the knowledge and skills they require. While tablets have become key educational tools in recent years, children still need to learn how to type on a computer. The design, development, and evaluation of a gamified application aimed at improving 10-finger typing skills are described in this work. This game is developed for Windows 64-Bit which it has balance the system computation power that are usually use for AAA game in the market. The development of this project are base on Game Development Life Cycle (GDLC) and basic typing mechanism. Blender 3D is used to create the game assets while Unity Engine 2020 is used to develop the game. This project is expected the players will be able to be improve their typing skill and exposed to ICT terms.

ABSTRAK

Projek ini memfokuskan pada pengembangan Vocab Defender: The Typing Shooting Game yang bertujuan untuk meningkatkan kemahiran menaip pemain dan mendedahkan pemain dengan istilah ICT yang digunakan dalam kehidupan seharian. Teknologi maklumat dan komunikasi (ICT) semakin banyak digunakan dalam persekitaran pendidikan untuk menawarkan pelajar pengetahuan dan kemahiran yang mereka perlukan. Walaupun tablet telah menjadi alat pendidikan utama dalam beberapa tahun kebelakangan ini, anak-anak masih perlu belajar cara menaip di komputer. Reka bentuk, pengembangan, dan penilaian aplikasi gamified yang bertujuan untuk meningkatkan kemahiran menaip 10 jari dijelaskan dalam karya ini. Permainan ini dikembangkan untuk Windows 64-Bit yang mempunyai keseimbangan kekuatan pengiraan sistem yang biasanya digunakan untuk permainan AAA di pasar. Pembangunan projek ini berdasarkan pada Game Development Life Cycle (GDLC) dan mekanisme menaip asas. Blender 3D digunakan untuk membuat aset permainan sementara Unity Engine 2020 digunakan untuk mengembangkan permainan. Projek ini diharapkan para pemain dapat meningkatkan kemahiran menaip mereka dan terdedah kepada istilah ICT.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
GUESS	**-**	**Game User Experience Satisfaction Scale**
UI	**-**	**User Interface**

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Project Background

Vocab Defender is an endless 3D shooting typing game and the theme chosen is digital world which is focusing on inside computer processor as virtual eyesight environment. Player acts as defender who needs to protect computer system by typing the Information Communication Technology (ICT) words or terms appear on the screen to all eliminate incoming viruses. As player finish typing each letters, a weapon will blast a nearest virus and player needs to keep typing all the IT terms on the screen to destroy all the viruses before damaging computer processor tower. All player need to do just fast typing skill in order to survive through this game. The purpose of Vocab Defender game development is to help players improve their typing skill beside expose ICT terms to players. This game will challenge player's typing skill while enjoying an interesting shooting animation and game environment. Players can save their result for each play round. They can track their progress of typing skill improvement by referring the scoreboard and aim to beat the highest score every time they play. Primary students to university/college students will be focused as the game target group.

1.2 Problem Statement

Our nation is targeting to become a developed country with most comprehensive use of technology. With the current Covid-19 situation happened, it force us and our country to take full advantage on the usage of technology. Most people right now need to do all work tasks and online meeting at their home. Students especially need to attend online class and complete their school tasks and

assignments online. Typing skill is very important in this technology era especially for students in completing their tasks in short time. So, the problem is, “Is students’ typing skill is good enough to help them completing their online school/university tasks or assignments?” And, “Are students get enough exposure to ICT terms and vocabulary?”. This project is developed on purpose to counter the problems above in improving players’ typing skills and give ICT terms exposure to them while enjoying the game world and game animation.

1.3 Objectives

i. To identify how typing game can improve typing skill and ICT terms or vocabulary exposure.
ii. To develop suitable and interesting typing game in helping typing skill improvement and ICT terms exposure.
iii. To evaluate the effectiveness of typing game developed in this project in improving typing skill and expose ICT terms.

1.4 Goals and Genre

The game goal is educational training and has entertainment element. The game’s main goal is to help players improve their typing skill while enjoying the game environment and animation. Besides that, other game goal is can give exposure of ICT terms that used in daily life among the players especially students.

The genre of Vocab Defender is 3D shooting typing game which the game mechanics is mainly and only typing keys on keyboard with the shooting animation and game world as the game fun factor.

1.5 Game Features

Entertainment Software Rating Board (ESRB) rating for Vocab Defender game is Everyone which is suitable for all age players because the game played using

simple typing mechanism and game content is free from heavy violence act which is just simple shooting and explosion animation.

The game rule is simple where player just need to defeat all the enemies (computer virus) by typing all the words appear on the screen and try to survive the endless enemy waves as many as possible. All player needs to do is just typing all the words on the screen to shoot the enemies. One typed letter is equal to one bullet and one bullet can kill one enemy. The more the enemy waves or rounds, the increase the number of enemies for each waves. The player's health bar will be an indicator of time limit which the game will over if the health bar is become 0% but there is an item which exist randomly throughout the game and need to be shoot to re-add the player's health. The health bar become empty can happen when enemies keep on shooting the core processor tower and make the player loose. So, player need to type fast to shoot all the enemies before the health bar is empty. The game result can be store if player has internet connection while playing the game and top 10 high score results can be viewed in scoreboard.

1.6 Conclusion

In conclusion, this chapter describes the overall idea about this project. The final product expected to be playable and fun which can improve players' typing skill and expose them with ICT terms. Typing game is one of best way to improve typing skill beside improve players' vocabulary in a fun way. Players can keep playing without feel bored easily because of the game itself is fun to be played. Moreover, most typing games require players to type out the displayed word while keeping their eyes on the screen. They encourage proper finger positioning in order to type quickly and efficiently.

In the next chapter, literature review will be elaborated and explained as well as the project methodology.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

The Covid-19 pandemic happened right now has left tremendous impact to all sectors all around the world. Education sector also not exception. Unplanned and drastic school cancellations may have a significant impact on students' academic interest and performance and digital learning need to be enhanced. Students will now forced to complete and submit all school/university tasks online. One of most useful skills for students right now is typing skills.

Enhancing typing skills will help students use current digital devices especially computers more efficiently to complete a variety of tasks. Typing software was once widely employed to improve students' typing skills, although it usually lacked an appealing multimedia impact. Games have grown in popularity in recent years, and one of the reasons is that they include a lot of appealing components, such as a lot of multimedia effects and unique screenplays. When it comes to the educational value of typing games, they are more engaging than traditional instruction, which is why individuals utilize them to improve their typing skills.

2.2 Genre

The genre for this game project is typing game. The typing game is a type of video game that mainly requires players to type. Typing games began as a subgenre of instructional games designed to educate players with keyboard use, but as players became more accustomed to using a keyboard and the games became more challenging and sophisticated, they evolved into their own category of games. A typing game would usually demand the player to swiftly or precisely type in words,

specific letters, numbers, or other keys that appear on the screen in order to progress in the game, serving as a challenge as well as a way to develop player's touch typing skills.

Furthermore, the subgenre for this game is shooting. Shooting game is when a weapon linked to the character's body allows the player to control the character and shoot targets (Hosch, 1978). For this game, player need to type all the words display on the screen and it will trigger the shooting action to destroy the enemies spawned in the game.

2.3 Existing Game

2.3.1 Introduction to Existing Games

1) TypeRacer

Figure 2.1 : TypeRacer

(https://play.typeracer.com/)

TypeRacer is a browser-based multiplayer typing game. Players compete against one another or with other internet users to complete typing tests of various texts as rapidly as possible in TypeRacer. It was first introduced in March of 2008. Alex Ephsteyn designed TypeRacer with the OpenSocial API and the Google Web Toolkit. Ephsteyn is a former Google intern who graduated from the University of Massachusetts Amherst with a master's degree in computer science. He was motivated to create a competitive multiplayer typing game after seeing that the

Windows shareware programme he used to learn touch typing didn't include a multiplayer mode. Despite the fact that other games like The Typing of the Dead had been launched previously to Ephsteyn's conception of TypeRacer, he was unaware of their existence due to his self-described lack of gaming skill.

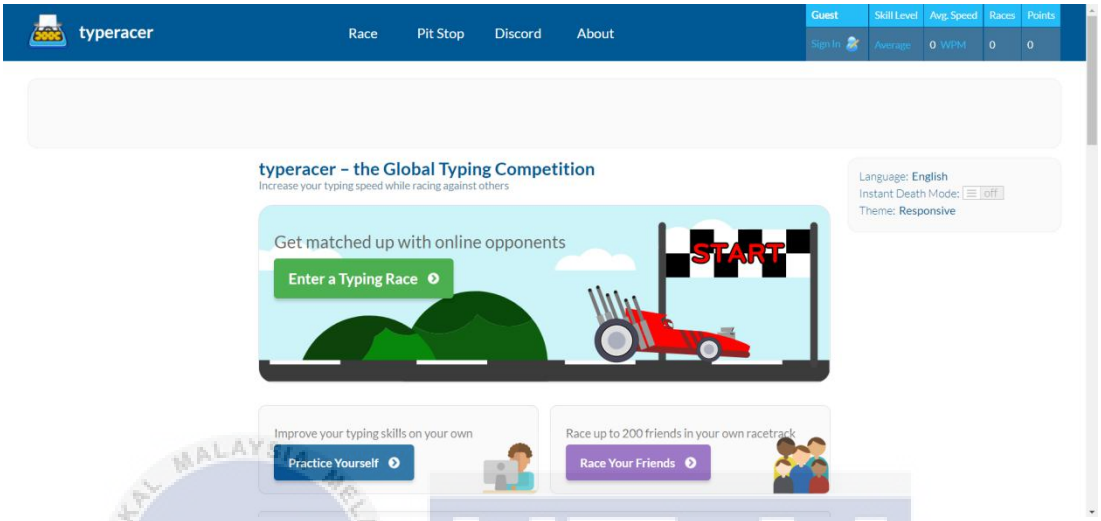

Figure 2.2 TypeRacer Website Main Page

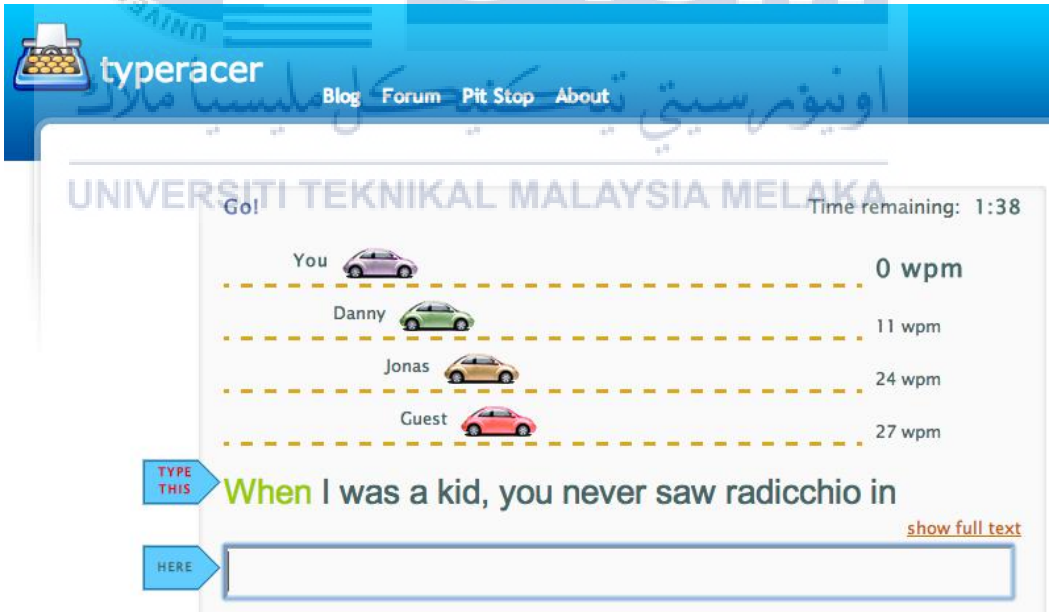

Figure 2.3 : TypeRacer Gameplay

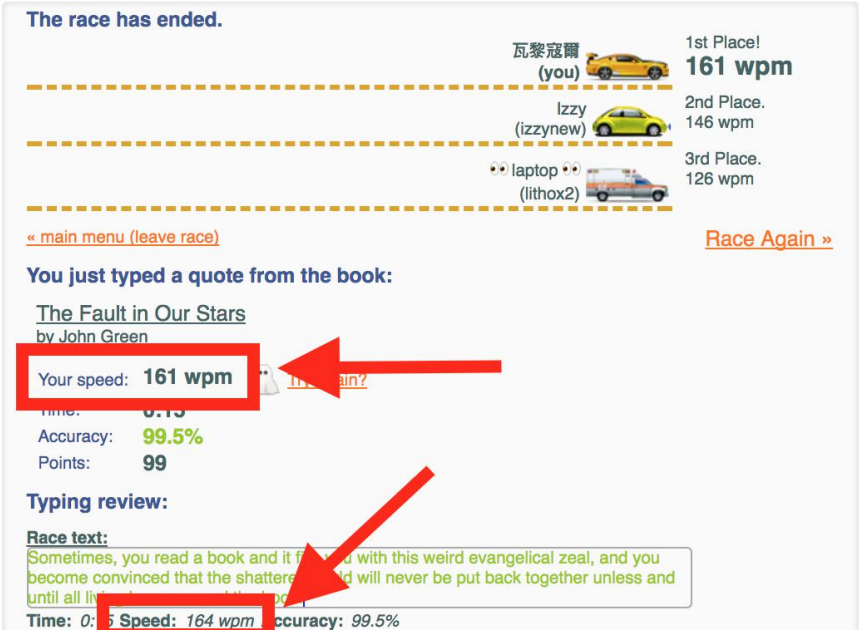

Figure 2.4 TypeRacer End Result

2) ZType Space Typing & Spelling (Android)

Figure 2.5 : ZType Space Typing & Spelling

Most hardcore gamers are familiar with Space Invaders, a 1970s game in which players used two-bullet lasers to shoot small green and purple aliens across a screen. Here's a game that goes above and beyond. ZType is a shooting game similar to Space Invaders, but with an added bonus: you may practise your keyboarding abilities while playing. ZType employs waves instead of aliens. To keep the ship from being blown up by missiles, the player must type certain terms. The longer and more complicated the words you must type within the time limit as you go in the game, the greater your score will be.

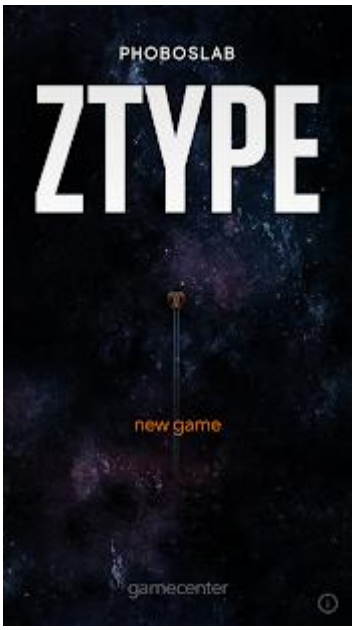

Figure 2.6 : ZType Start Menu

Figure 2.7 : ZType Gameplay

Every letter had typed will fire a strong plasma bullet at the enemy spacecraft, so better type fast!

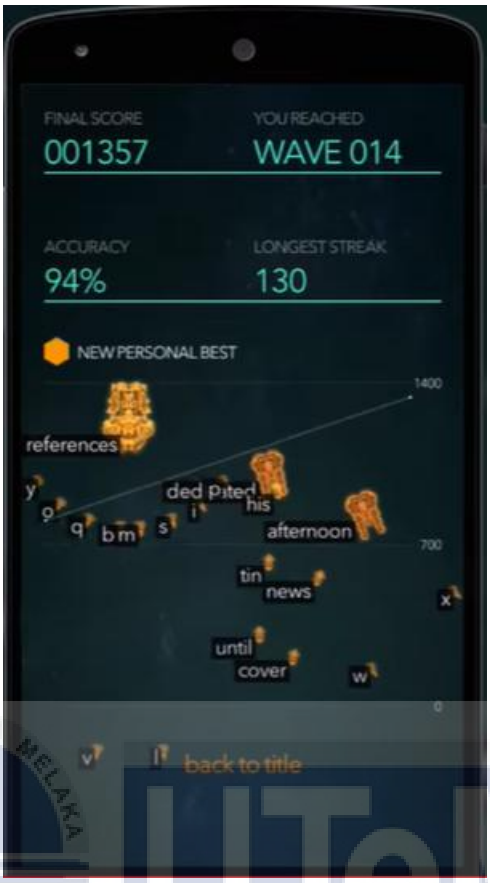

Figure 2.8 : ZType Result Layout

2.3.2 Comparison of Existing Games

	TypeRacer	ZType Space Typing & Spelling
Gameplay	● Several players compete by racing 2D car that move forward as the players type different sections. The faster player types, the faster car can move forward ● The paragraphs are anything between 20 and 930 characters long are needed to be typed as	● Played only by single player ● Player types all the words display on the screen to blast all the alien spacecrafts. Player needs to survive as long as he/she can.

	in sequence. ● There are several game modes which are maintrack (default), practice, competitions and private tracks. Most of them are Multiplayer except practice mode.	● One alien spacecraft has one word. All the words spawn randomly follow the spacecraft and will go straight to player. ● Once one letter complete typed, one plasma bullet is fired to the selected alien spacecraft.
Game Mechanics	● Using typing mechanism on physical keyboard ● Involve all letter, number and symbol keys ● any typing errors detected in spelling, capitalization or punctuation must be fixed by the player before continuing with the race.	● Using typing mechanism on virtual keyboard displayed on smartphone screen ● Consists of QWERTZ, AZERTY, Colemak and Dvorak Keyboard Layouts that player can choose ● Involve only all letter keys and using only lower case letters
Platform	HTML5 support browser with a physical keyboard	Compatible for Android and iOS
Duration	Depending on text length	As long as player can survive

Table 2.1 : Comparison of Existing Games

2.4 Project Methodology

Several type or version of Game Development Life Cycle (GDLC) had created for game development in order to overcome some challenges and to replace Software Development Life Cycle (SDLC) (Rido Ramadan and Yani Widyani, 2013). Heather Chandler GDLC is chosen to be used as the project methodology in this project.

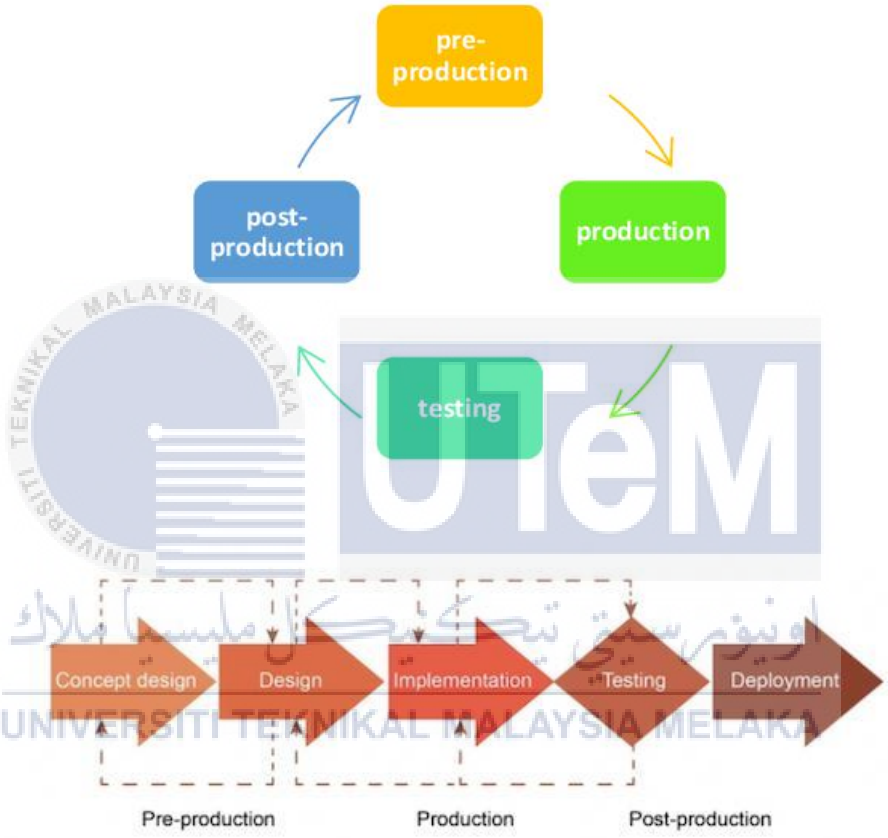

Figure 2.9 & 2.10 : Heather Chandler GDLC

Pre-production

1. Concept Design

The process of generating new ideas through brainstorming. All random ideas and topic matter are accounted for as a previous notion. What is the problem statement, how do you solve it, how should the game appear, and how much does it cost in terms of time and resources. Early design process is being done in this phase.

Production

2. Design

The method by which core design is carried out. All design concepts are visualised as a more concrete product, either on paper or as a prototype. World design, character design, and other game models will all be part of the design process.

3. Implementation

Source code and programming are both involved in this process. Physics development, game mechanics, data management, and game logic source code will all be part of the implementation. By the end of this phase, the game should be playable.

Post-production

4. Testing

The process of putting the entire game through its paces in order to improve it. It is divided into two phases, which are referred to as alpha and beta. During the alpha test phase, the game is tested by a small group of people or the development team itself, whereas during the beta test phase, the game is tested by a focus group or a specific target user to obtain information and input to improve the game. In this situation, the programmer will test the alpha version, while the beta version will be tested during the demonstration.

5. Deployment

Where the whole complete version of the game, often known as the master version, is released to the market for public sale or endorsement as a free downloading game.

2.5 Conclusion

In the nutshell, this chapter discussed about the reason why certain type of genre is being chose. Other than that, the introduction of existing games and their comparison being explained where the game that will be developed has similarity with those existing games and suitable project methodology also being discussed and how it will be implemented in this project by using GDLC.

For the next chapter, requirement analysis that consists of project requirement, technical requirement, software requirement, hardware requirement and other requirement if applicable will be discussed along with the project schedule and milestone.

CHAPTER 3: ANALYSIS

3.1 Introduction

Based on the the existing similar games and their differences that had been mentioned in Chapter 2, the analysis phase and the game development process will be discuss in deep in this chapter. We will go through and analyze the game components and game features in the part of player roles, gameplay, victory condition, core mechanic, level progression and user interface.

3.2 Requirement Analysis

3.2.1 Project Requirement

Player Roles

TypeRacer: The player need to rush the 2D car forward by typing the text displayed on screen. The faster player types, the faster car move forward.

ZType: The player need to blast all the incoming alien spacecraft by typing random spawned words appear on the screen.

Gameplay

TypeRacer: Several players compete in who is the fastest 2D racing car by typing fast all the words in the text essay.

ZType: Single player need to survive destroying all the alien spacecraft by typing all the random words appear on screen. The many and faster player types, the longer player can survive.

Victory Condition

TypeRacer: To win a round of the game, player must type faster than other players and the player's car arrive the finish line first.

ZType: Player just needs to survive the game as long as and as many waves as player can.

Core Mechanic

TypeRacer: The mechanic in this game is simple, player just focus on typing all the words using the keyboard. Whoever type the fastest, he/she is the winner.

ZType: Same as TypeRacer, the game mechanic is simple, player just need to concentrate typing all appeared random words to survive the game.

Level Progression

TypeRacer: If player already sign up in the game website, the result will be stored automatically and it will determine player's experience level which is called Typist 0 to Typist 9 (basic account) and Racer 0 to Racer 9 (premium account). The many and good player plays, the experience level will increase.

ZType: This game using a wave progression. As the wave increase, the alien spacecraft and the random words that are spawned also will increase and move faster to player spaceship for each waves. Once game over, the game will start from first wave if player play again. The end result will be stored automatically in the game application as score graph after game over. Player can see the score graph after each game over.

User Interface

TypeRacer: The user interface is not too compact and not too simple, can be understood and easy to navigate. Players can choose the game mode they want on above side of the game website and scroll down to see the high score. In-game interface quite simple which is using 2D sprites and graphics. There are only several 2D car on track and below it the text that need to be typed. After game finished, the complete game result will appear below the text.

ZType: User interface for this game is simple. For the main menu, there is only game title, player's 2D spaceship, 'new game' button and 'setting' button with beautiful outer space background. When game start, virtual keyboard and small EMP number are displayed. The complete game result only appear after game is over.

3.2.2 Technical Requirement

Keyboard and Mouse

Traditional input devices for computer gaming, such as the mouse and keyboard, have been available since the beginning of the industry. In gaming, the mouse and keyboard enable better dexterity and precision. More hotkeys may be managed with a keyboard, making it easier for the player to change appropriate input to fit their tastes. With today's modern gaming mice, players can also employ some of the mouse's built-in hotkeys. Gaming mice now available in a wide range of designs and functions, including DPI speed control, weight control, wireless functionality, and RGB colour. Mechanical keys, wireless connectivity, and RGB colour options are all available on gaming keyboards.

Game Engine

This project is using Unity Engine version 2020.3.8f1 for the game development. The Unity game engine is a leading and one of the most popular platforms for game developers, because it is both free to use and royalty-free. Unity allows users to create 2D and 3D games and experiences on a variety of platforms.

The Unity editor and games both have a C# scripting API, as well as drag-and-drop functionality. Unity supports the loading of sprites and includes a robust 2D environment renderer for 2D games. For 3D games, Unity supports bump mapping, reflection mapping, parallax mapping, SSAO (screen space ambient occlusion), dynamic shadows with shadow maps, render-to-texture, and full-screen post-processing effects.

3.2.2.1 Software Requirement

Software required for this project are listed below:

- Unity 2020.3.8f1 (Game Engine)
- Audacity 2.4.1 (Audio tool)
- Blender 2.90 (3D model editor)
- Mendeley 1.19.8 (Reference manager)
- Microsoft Word 2016 (Word processor)
- Google Chrome 91.0.4472.77 (Internet browser)

3.2.2.2 Hardware Requirement

Hardware required to develop this project and for player interaction are:

- Laptop / PC
- Mouse and Keyboard
- Speaker or Headphone or Earphone
- Internet connection

3.2.2.3 Other Requirement

These are required tools or platform required in assisting the project development during the Movement and Control Order (MCO):

- Google Meet (Online meeting platform)
- Google Drive (Cloud Backup)
- UTeM Official Learning Management System (ULEARN) (E-learning portal)

3.3 Project Schedule and Milestone

Project schedule and milestone is important to ensure and help the development process to finish on the intended duration (Table 3.1).

Key Milestone	Start Date	End Date
Concept & Idea	15 March 2021	4 April 2021
Report (Chapter 1 & 2)	22 March 2021	4 April 2021
Design	29 March 2021	23 May 2021
Report (Chapter 3)	5 April 2021	18 April 2021
Implementation	29 March 2021	21 June 2021
Report (Chapter 4)	12 April 2021	25 April 2021
Testing	31 May 2021	21 June 2021
Deployment/Presentation	16 June 2021	22 June 2021

Table 3.1: Project Schedule and Milestone

Table 3.2: Development and Report Gantt Chart

Gantt chart for the project helps developer to visibly see the schedule timetable to meet the intended duration easier (Table 3.2).

3.4 Conclusion

In the conclusion, the topics of requirement analysis, project timeline, and milestones have been covered in this chapter. The requirement analysis includes the differences between similar games that are relevant to this project. Then, in addition to reviewing the project's hardware requirements, we reviewed the project's technical requirements as well as software requirements. We also discussed the various requirements that have arisen as a result of the current pandemic crisis. The project schedule and milestone planning, which depicts the stages of project development, were also discussed.

In the next chapter, we will be discussing about the design of the project development.

CHAPTER 4: DESIGN

4.1 Introduction

In this chapter, we will talk about game design, which is an important aspect of the game creation process. The game architecture, including how the game works, gameplay, challenge hierarchy, core mechanics, flowboard, level progression, storyline, user interface, game arts, game world, character design, camera model, audio and sound effect, and all the planning and procedures that have been carried out, will all be explained.

4.2 Game Architecture

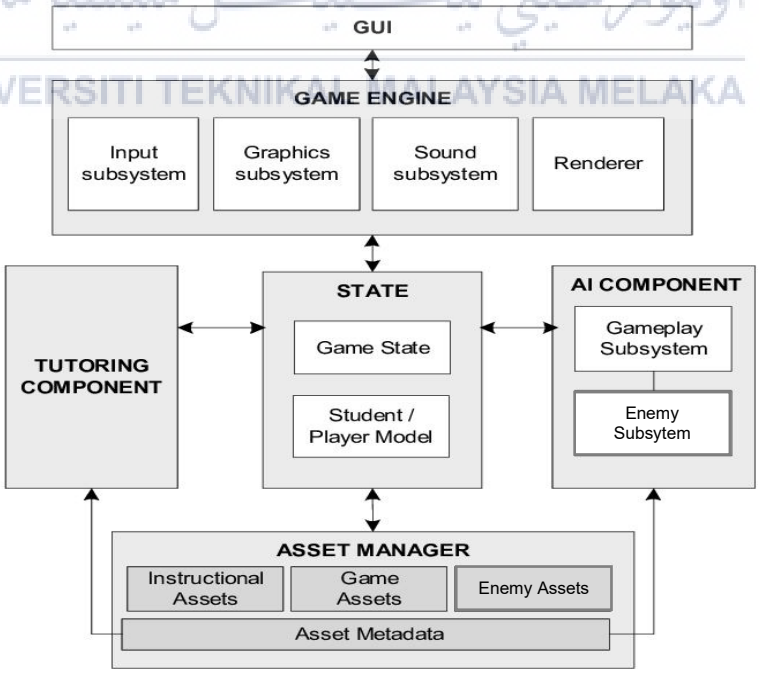

Figure 4.1: Game Architecture Diagram

4.3 Game Design

4.3.1 Gameplay

- **Player Roles**: Player just need to focus on typing fast all the words appear on the screen

- **Game Rules**:

  i. Player need to type fast to destroy and prevent enemy attack/shoot core tower.

  ii. Player just can type one word at a time because only one word appear at one time. Next word appears after current word finish typed.

  iii. One letter equal to one bullet. Bullet will be shot to enemy after correct letter is typed.

  iv. Enemy will destroy after get hit by 3 bullets. Next enemy wave will come after all enemies in current wave are killed.

  v. Player's health will be deducted if enemy shoot tower or enemy collide with the tower and game is over after player's health reach 0%

  vi. As the wave number increase, the number of enemy also increase. Player need to survive as long as they can. The end result will be displayed after game over/player loose.

- **Victory Conditions :** Players can survive and defending tower as long as they can

- **Level of Difficulty :** As the wave number increase, the number of enemy also increase.

4.3.2 Core Mechanics

- Player movement : Tower's turret blaster detect the nearest enemy automatically and will blast the enemy after player typed correct letters follow the letter shown in the screen

- Virus1: Enemy that can move around the core tower and has the lowest fire rate

- Virus2: Enemy that can move around the core tower and has fire rate faster than Virus1

- Virus3: Enemy that will go straight to the core tower and collide with it

- Virus4: Static enemy that has highest fire rate

- Save & Load System: After game over, the end result will be stored into PlayFab system automatically if player has internet connection while play the game.

- Collectible Items : Health bar will fill up if player shoot the green medic core which spawn randomly.

- Main Menu : menu that have Play button, Scoreboard button, and Quit button

4.3.3 Flowboard

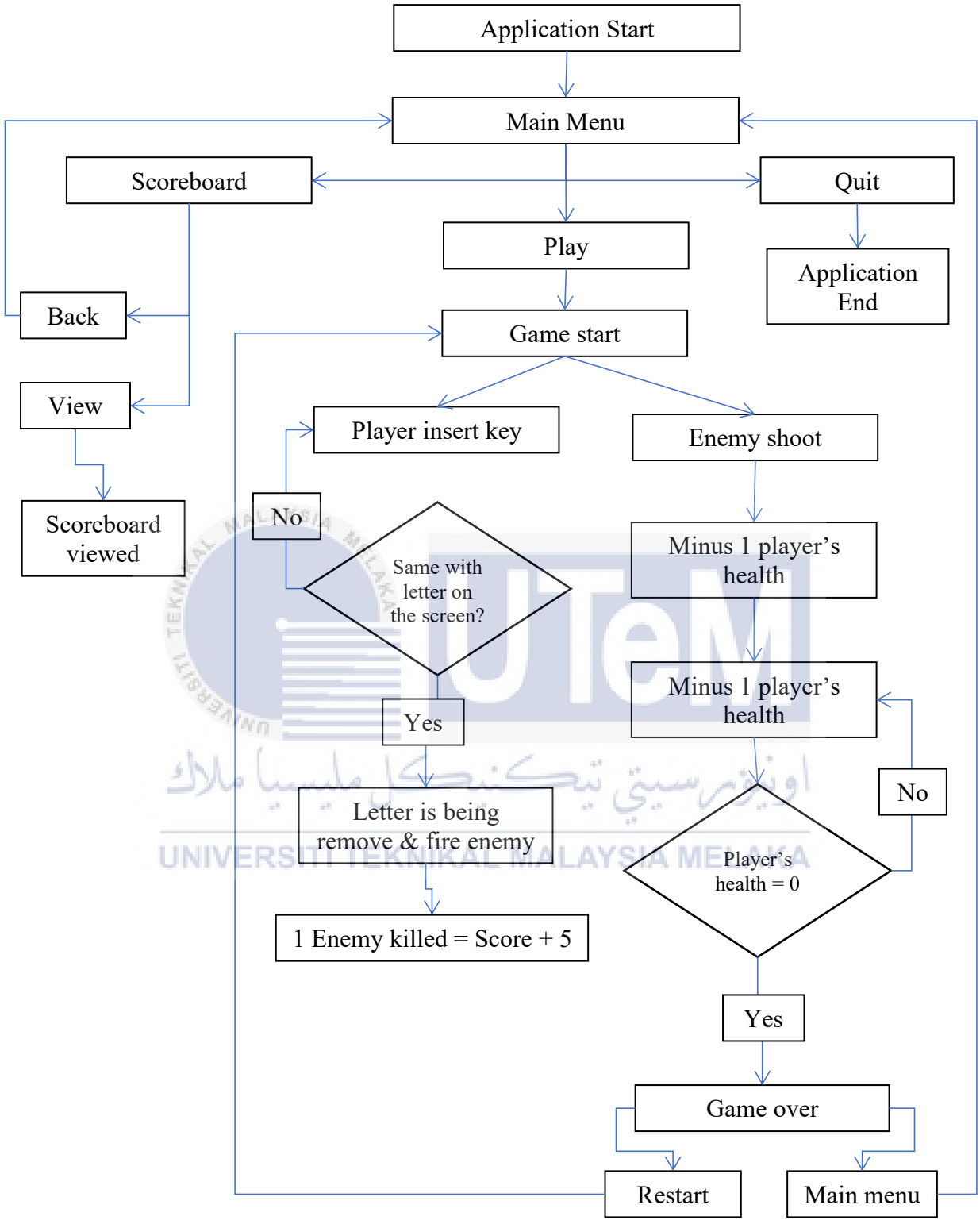

Figure 4.2 Flowboard Vocab Defender

4.3.4 Level Progression

Player can review their previous game result at scoreboard layout. Player can see their progress on typing skill and try to beat the highest score on the next game play.

4.3.5 Storyline

The story behind this game is there several viruses had affected some parts in computer system. Now, the viruses come to the last section which is computer processor. It will over if last defense has fallen. So, it is a job for typist to defend the last section from get affected by the incoming viruses.

4.3.6 User Interface (UI)

Vocab Defender main menu UI has 3 clickable buttons (Figure 4.3). Click 'PLAY' button to start the game, 'SCOREBOARD' button to view scoreboard and 'QUIT' button to exit the game application.

Figure 4.3: Main Menu UI

In-game UI in Figure 4.4 consists of time in seconds, time interval for each wave, score, the number of word, words per minute (wpm), accuracy in % and player's health bar. There are 8 buttons which player can enable 8 turrets when

player reach certain wave/round. Main UI is word that need to be typed is display at a right font size and position so that player can see and type correct letters.

Figure 4.4: In-game UI

Player needs to prevent the health bar reach 0% caused by the enemies' attack and give damage to the core tower. When the health bar reach 0%, it will be a game over and game result UI (Figure 4.5) will appear. This UI will show the score, the number of word that been typed, the round that player reached, typing accuracy and typing speed in WPM.

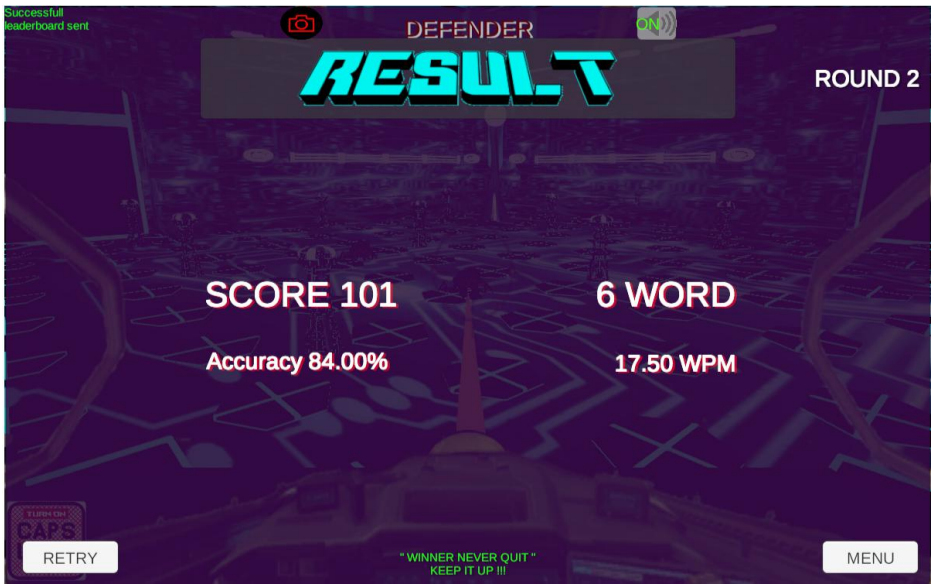

Figure 4.5: Game Result UI

In Scoreboard layout, there will be 2 button which are ‘BACK’ button for go back to main menu and ‘VIEW’ button to display current scoreboard which listed according to the highest score.

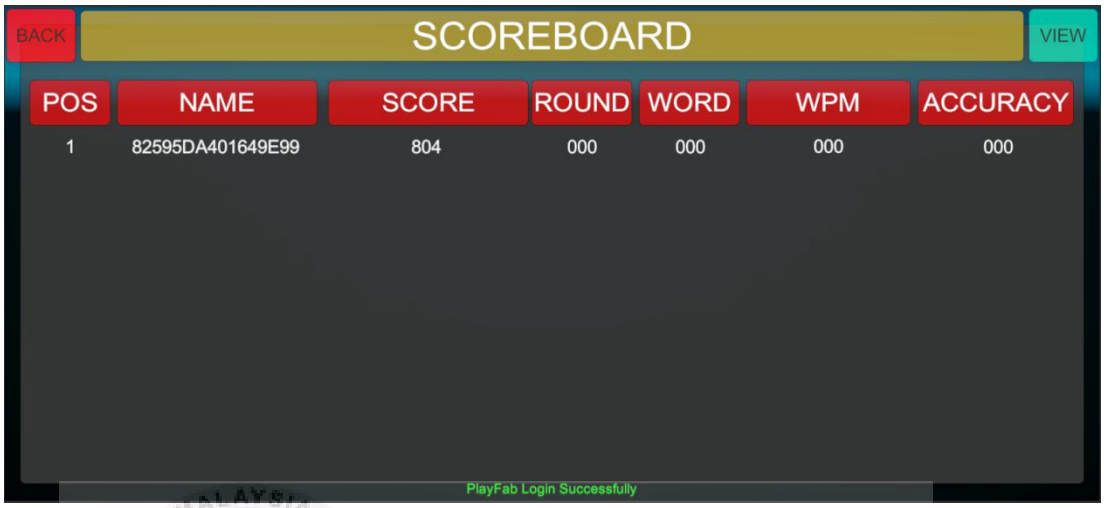

Figure 4.6 : Scoreboard UI

4.4 Game Art

4.4.1 Character Design

1) Main Proccesor Core Tower

- The main tower that need to be protected from viruses attack

Figure 4.7 : Main Processor Core Tower

2) Turret Blaster

- Main weapon attached to the main tower. While detect and tag the nearest virus.

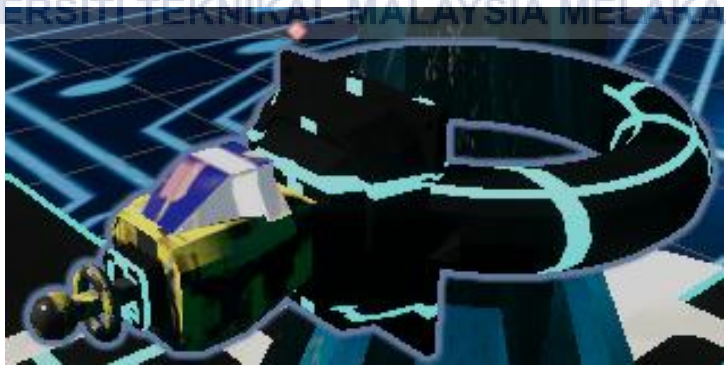

Figure 4.8 : Turret Blaster

3) Side Core Tower

- Generate power to main tower. Viruses can also attack side tower to deduct the main tower's health.

Figure 4.9 : Side Core Tower

4) Viruses

- Computer virus that come and attack all the towers inside the processor

Figure 4.10 : Viruses (enemy)

4.4.2 Game World

1) Processor Last Defense

- Inside computer digital view, the event place is at the computer chip core processor on the motherboard.

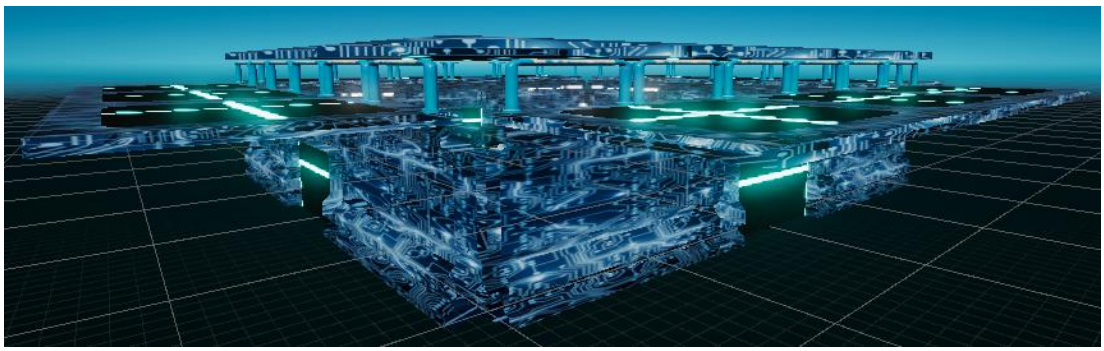

Figure 4.11 : Outside view Processor Last Defense

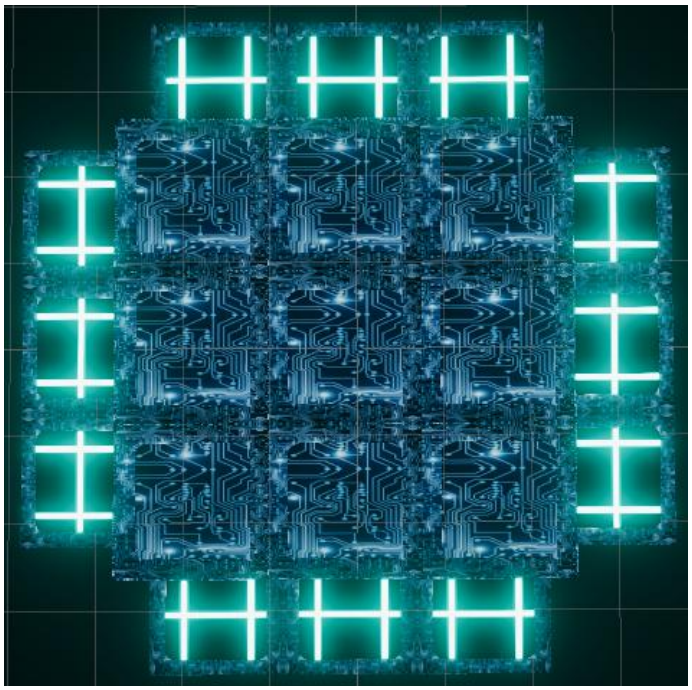

Figure 4.12 Top view Processor Last Defense

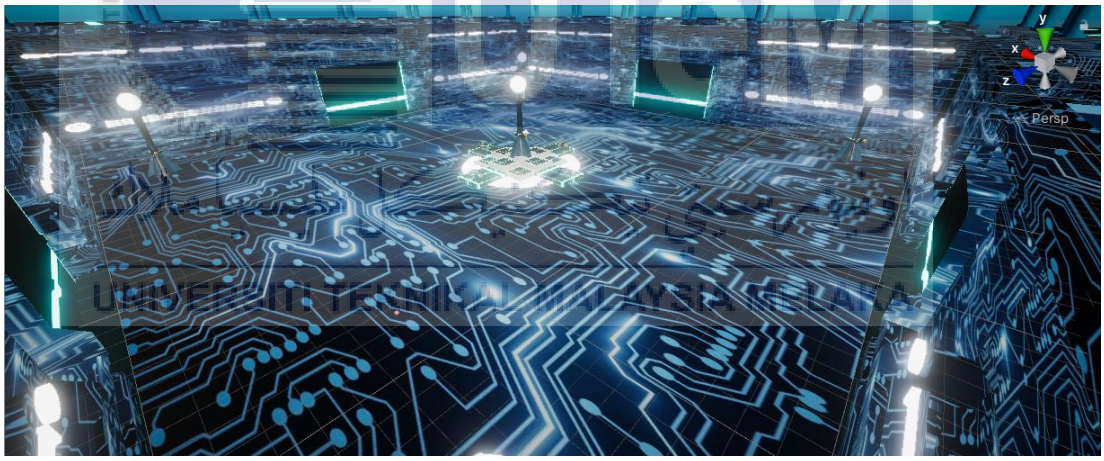

Figure 4.13 Inside view Processor Last Defense

4.4.3 Camera Model

The camera perspective is from 3D first-person view. The camera is attached to the turret at main core tower. The turret is set to detect and follow the nearest enemy direction automatically. The main camera will follow the turret movement rotate 360 degree on y-axis.

Figure 4.14: First-Person view

4.4.4 Audio/Sound Effect

Audio used in the game is mainly from royalty free sites, with a few that is recorded, modified or tuned using Audacity. The background music is mainly use like electronic genre which is suitable with the game theme.

Other than that, sound effects also applied in the game which covers the shooting, enemy's explosion and button click.

4.5 Conclusion

In conclusion, all contents of the game architecture, overall design of the games, including gameplay, hierarchy of challenges, core mechanics, flowboard, level progression, storyline, user interface, game arts, game world, character design, camera model, and audio sound effects, have all been explained in this chapter. Details and descriptions were explained, and concepts and sketches were converted into digital form and included into the game project.

In the next chapter, we will discuss about how the game is implemented, including the creation of game art, visuals, audio, and animations, game component integration, game configuration management, configuration setup, version control procedure, and implementation status.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, we will discuss in detail about one of important phase in project development process which is implementation. Everything that was designed in Chapter IV will be implemented in this chapter. This chapter will cover the technical aspects of implementation from the standpoint of a programmer throughout the production of graphics, audio, video, and animation. Other than that, we will also go over the media integration, configuration management, and implementation status.

5.2 Media Creation

5.2.1 Production of Texts

In this game, text is used to describe the game title, the words appear on the main game screen, all information like the number of score and round that player reached and also on some buttons in user interface like main menu and game result UI. As a result, in Figure 5.1, there is an example of text for game title using Nazalization font, and in Figure 5.2, the result text is using Satisfontory font which both available from fontmeme.com while other texts used in the game (Figure 5.3) are using Arial font which is a default font used in Unity Engine. The font size for each text varies based on the function of the text. The game title and game result title will be aligned to the centre of the game screen, and the font size will be larger than usual. Furthermore, each text will have an outline to give it a 3D dimension aspect.

Figure 5.1 : Game Title text

Figure 5.2 : Game Result text

Figure 5.3 : Example of other texts used in the game

5.2.2 Production of Graphics

The first phase of graphics production is producing a 3D character model and assets in Blender 3D which include modeling, texturing, rigging, and animating. The 3D models are then loaded into Unity Engine and made available for use. The game world is generated inside the game engine utilizing terrain tools, UI tools, lighting tools, post-processing, and so on in the second phase.

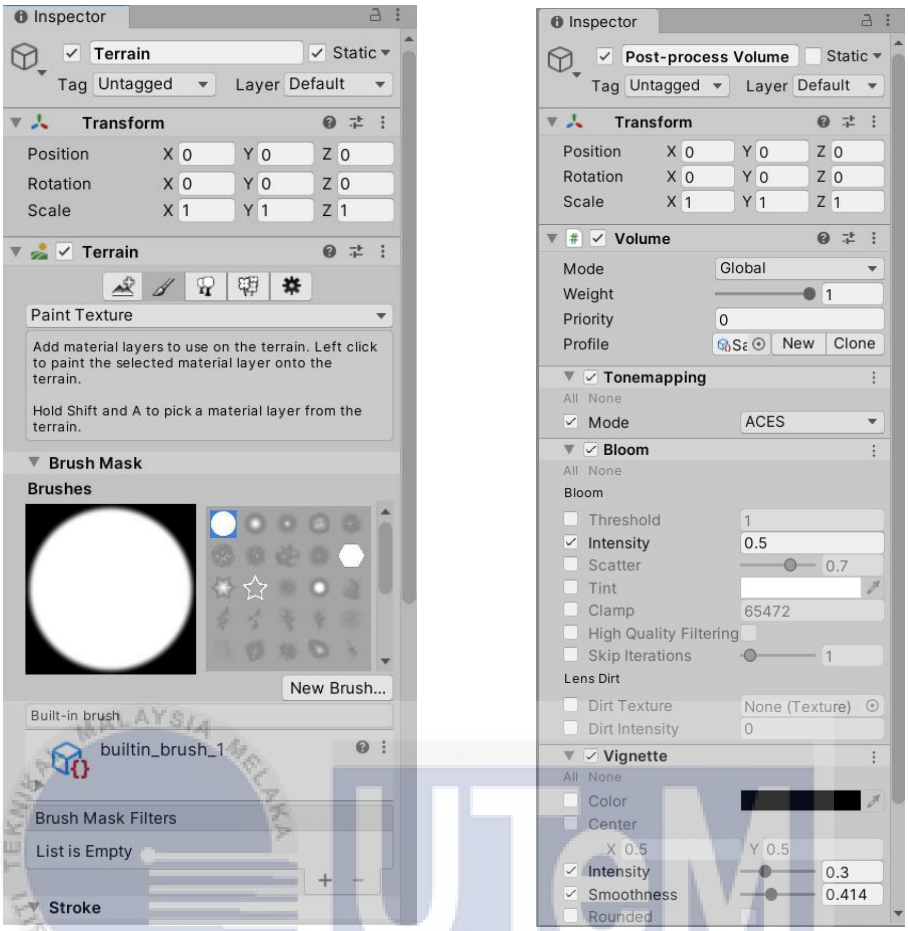

Figure 5.4 : Terrain tool and Post-process

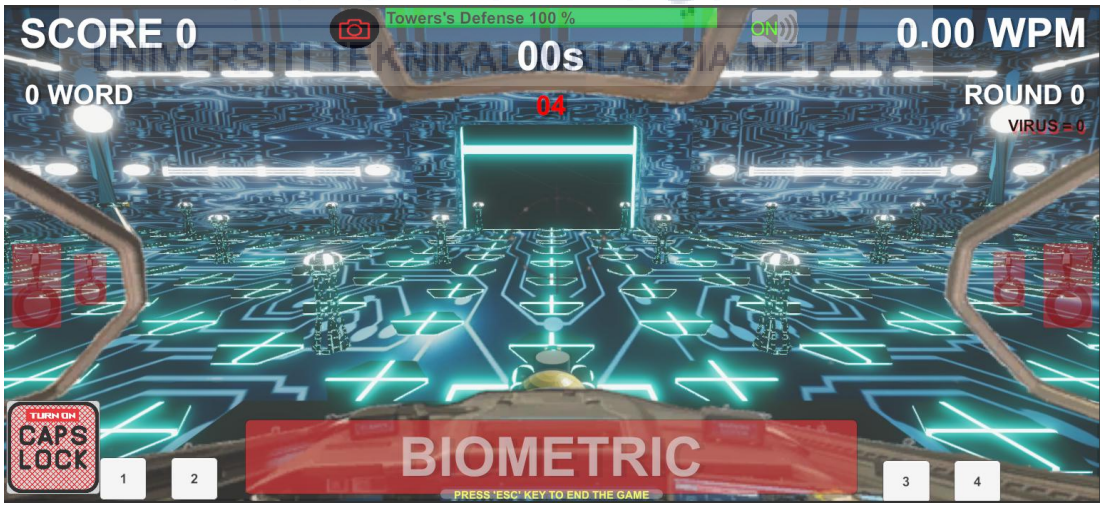

Figure 5.5 : Example of main game graphics

5.2.3 Production of Audio

The background music, enemies' sound, laser gunshot sound and correct and miss typing sound are the several audios used in this project. The audio comes from bensound.com, which provides royalty-free audio. The audio is converted to an mp3 file and imported into Unity, where it will be used throughout the project.

After the audio file being imported, we are allowed to configure the setting for the sound (Figure 5.6). The audios like enemies' sound, laser gunshot and others were controlled or triggered by some lines of coding. Figure 5.7 shows some coding lines to trigger audios like background music or error sound when player miss to insert a correct letter.

Figure 5.6 : Audio Configuration

```
if (nummusic == 0)
{
    music1.Play();
    music2.Stop();
    music3.Stop();
    music4.Stop();
    music5.Stop();
    musicoff.text = "ON";
    musicoff.color = Color.green;
}
```

```
else if (!IsCorrectLetter(typedLetter))
{
    errorinput.Play();
```

Figure 5.7 : Coding lines to trigger background music and error sound

5.2.4 Production of Video

There is no video implementation inside the game.

5.2.5 Production of Animation

In Unity Engine, to play or manipulate the 3d characters or assets animation is by using animator controller. For example, the enemies' animation state like idle, walk and jump is changeable by using animator controller (Figure 5.8) and then implement in the enemy's inspector (Figure 5.9).

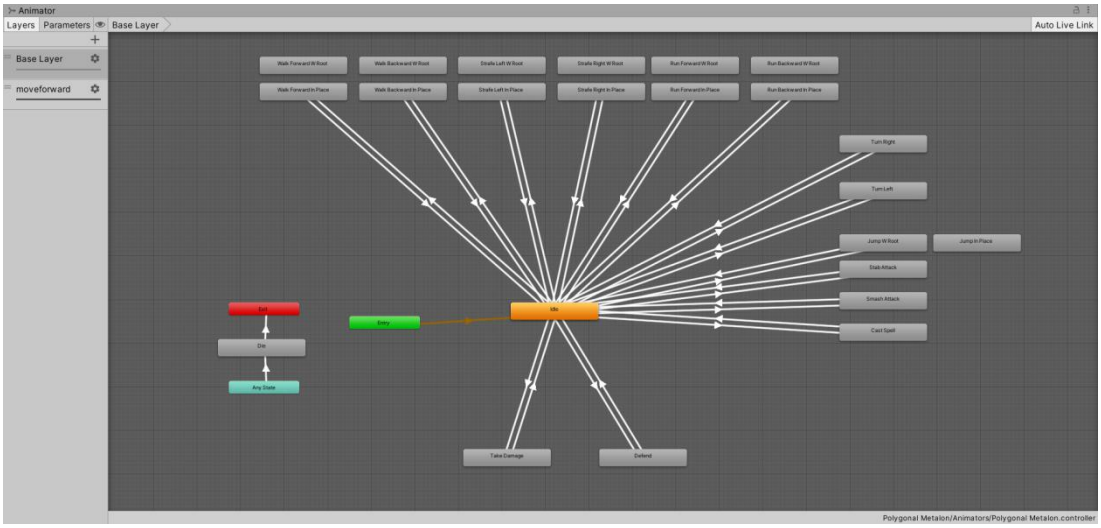

Figure 5.8 : Animator controller for enemies' animation

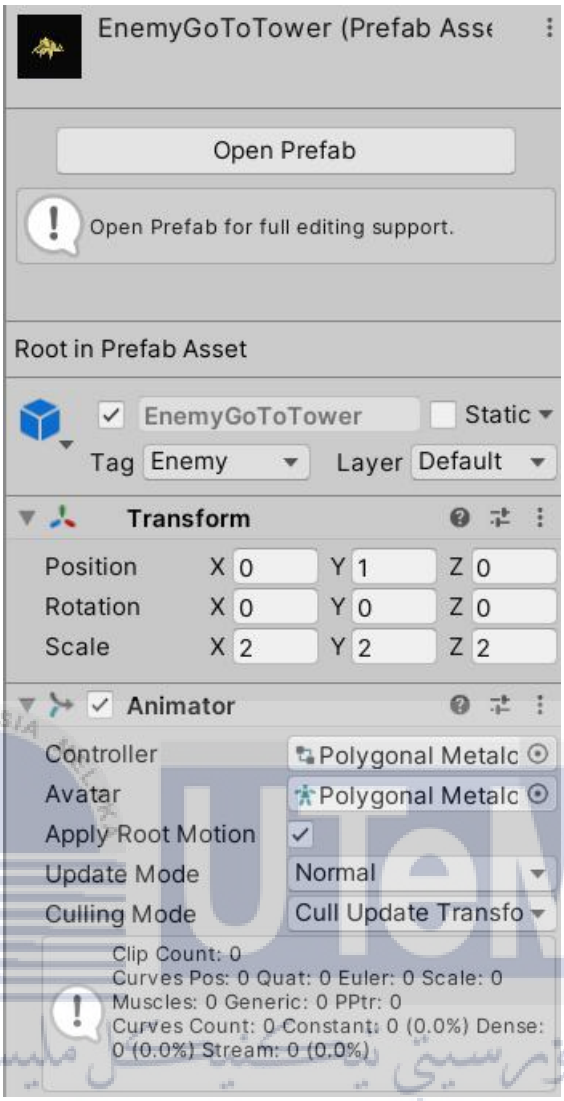

Figure 5.9 : Animator can be manipulated

For player's animation, a script is written to manipulate player behaviour like find the nearest target by rotating and shoot the target (Figure 5.10).

```
//Target lock on
Vector3 dir = target.position - transform.position;
Quaternion lookRotation = Quaternion.LookRotation(dir);
Vector3 rotation = Quaternion.Lerp(partToRotate.rotation, lookRotation, Time.deltaTime * turnSpeed).eulerAngles;
partToRotate.rotation = Quaternion.Euler(rotation.x, rotation.y, rotation.z);
```

```
public void Shoot()
{

    Debug.Log("SHOOT!");
    GameObject bulletGO = (GameObject)Instantiate(bulletPrefab, firePoint.position, firePoint.rotation);
    Bullet bullet = bulletGO.GetComponent<Bullet>();
```

Figure 5.10 : Some coding line for player's behaviour

5.3 Media Integration

As shown below in Figure 5.11, this is the example of media integration in this project. Media integration is critical in the game development process because it enhances the user's experience while playing the game.

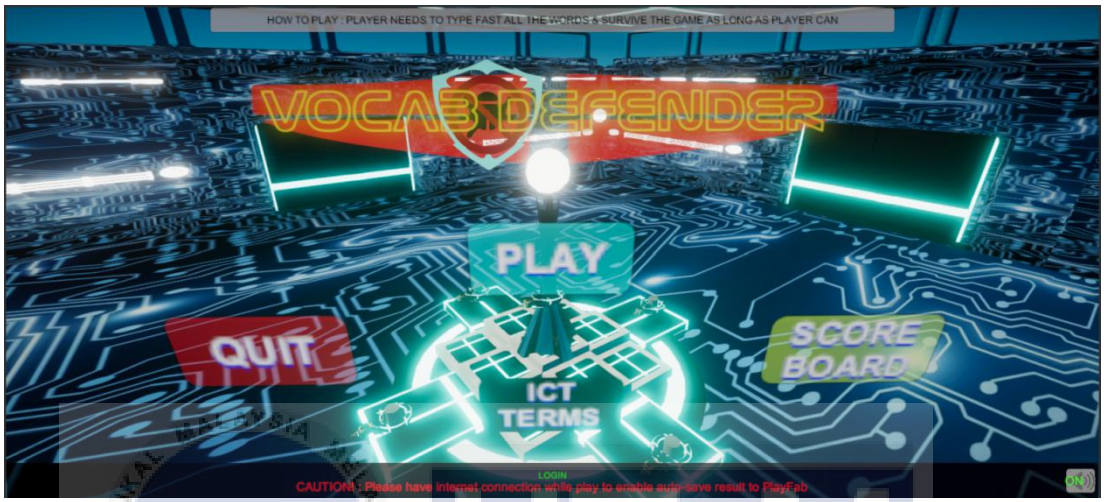

Figure 5.11 : Example of media integration in the game

To begin, the developer will design each scene required for this project, including text, graphics, music, and animation, in order to merge all of the elements into a single screen. After that, the developer will choose an appropriate theme for this game. This serves as a guide for the creator as well as adding to the game's fantasy element. The developer then reduces the use of strong colours throughout the scene.

Aside from that, the music background plays throughout the game and will loop if the player continues to play the game, and each button produces a clicking sound in order to provide some feedback interaction to the player, including the shooting buttons (all alphabet, symbols and numbers button) each time player types the words appear on the screen in this game.

As shown below in Figure 5.12 is another example of media integration. As the game start, a group of enemies will walk in and player's turret will rotate as detect the nearest enemy. This scene will always play the music background and can

be changed or turn off by clicking the speaker icon. The whole game assets are arranged and implemented as the game graphic is created using Unity 3D.

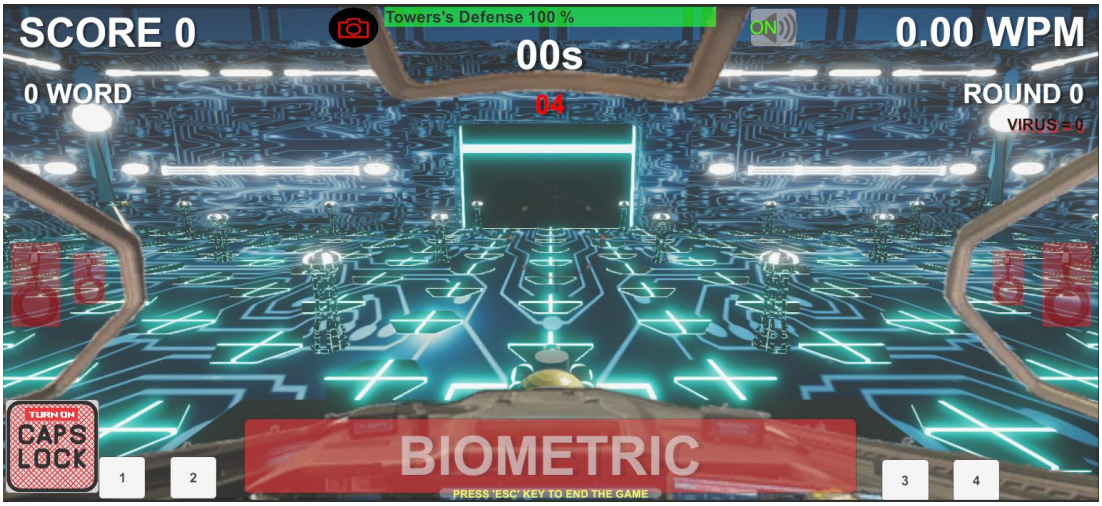

Figure 5.12 : Main game graphics

5.4 Product Configuration Management

Product configuration management is critical in game development since it is a procedure for ensuring and maintaining the game's consistency, performance, and functionality. As a result, this project goes through numerous product configuration management processes to ensure that it works as it should and that it is defined and documented in sufficient depth to meet its expected life cycle. As a result, this section will go through all of the information concerning the game engine, plug-in libraries, and so on.

Configuration Environment Setup

First, as shown in Figure 5.13, the complete project was constructed with Unity version 2020.3.8, which includes several enhancements over the previous version. The new Unity version is also more stable than the old one. It necessitates a few setups inside the project settings, such as setting up the map, which is designed to be the player's initial engagement with the app, building lighting based on quality, and putting up other important game information, such as the project title, logo, and

so on. Next, the PlayFab plugin is utilised to swiftly generate leaderboards in order to engage players more fully. Players can use their previous score as a benchmark for improving their performance in subsequent games.

Blender 2.91.0 is used for 3d modelling. It is the fourth major release in 2020, featuring powerful new booleans, improved cloth sculpting with collision support, volume object modifiers, updated animation tools, and much more.

Figure 5.13 : Unity Engine with plugin Playfab and Blender

Version Control Procedure

Testing Phase	Procedure	Control
Alpha	Programmers will test the game to see what works and what doesn't, as well as major issues that influence the game in major ways.	In form of early prototype and must be at least playable. The majority of the time will be spent on mechanics and localising content. This will be implemented during Project 1 in the final year.
Beta	The game will be tested by a target focus group, and feedback and questionnaires will be collected from them in order to improve the game further.	Following the evaluator's and supervisor's feedback during the Final Year Project 1 presentation. The game will enter a semi-Beta phase, during which it will be improved in terms of crucial aspects such as the game's goal. Then, during Final Year Project 2, it will reach the final phase, where input from the target focus group will be obtained.
Golden Version	The final version of the game, which is ready to be released on the market.	After evaluation in Final Year Project 2

Table 5.1 : Testing Phases

5.5 Implementation Status

By narrowing down the phase in the Gantt Chart and Milestone Table, we will be focusing on the specific components during implementation phase.

Component	Description	Duration to Complete	Completed Duration	Status
Character and Asset Creation	Modelling, Texturing, Rigging and Animating 3D models inside Blender	1 Week	1 Week	On Time
Game World Creation	Design the terrain according to the game theme and painting different type of textures	2 Weeks	2 Weeks	On Time
Interface Elements and Implementation	Designing game user interfaces and user interfaces in the game engine using 2D and 3D elements	2 Weeks	12 Days	In Time
Game Mechanic and Programming Implementation	Implement all the functions and mechanics inside the game from the gameplay element until interface Functionality. The testing and bug fix also be done together	6 Weeks	8 Weeks	Delay

Animation Management and Implementation	The animation will be set and implemented with source code and conditions inside animation for ready use	1 Week	1 Weeks	On Time
Texturing and Lighting Build	To build the textures and lighting in the game since editor only set for preview mode in order to give faster computation time.	1 Week	3 Days	In Time
Polishing All Aspects	Polishing any leftover or small details for final touch including project settings that need to be configure before exporting. Final testing and bug/error fix also be done.	1 Week	1 Week	On Time

Table 5.2 : Gantt Chart of Implementation Status

5.6 Conclusion

Overall, this chapter is related to the previous chapter which is design. The implementation phase is important for the game mechanics to work and be playable beside to make sure that the project will complete within the dateline. It encompasses the creation of game art and model using Blender, which includes graphics, audio, video, and animation. In terms of game integration, C# was used in this project, which was built with Unity Engine 2020.3.8 and Visual Studio 2019.

The next chapter will cover the game's usability testing, including the testing plan, technique, and data analysis.

CHAPTER 6: TESTING

6.1 Introduction

The fourth phase of the Game Development Life Cycle is testing, which is critical for evaluating the game's usefulness and correcting or fixing any problems, exploits, or malfunctions that may arise. We will concentrate on usability testing in this project, which is a technique used in user-centered interaction design to evaluate a product by testing it on players. The goal of this project is to assess the efficiency of the typing game produced in developing typing skills and to expose target users to ICT concepts based on the project's problem statement and requirements. The test plan, test strategy, test implementation, test results, and analysis will all be covered in this chapter.

6.2 Test Plan

The purpose of this testing is to determine how well the game is received by the players in terms of increasing their typing skills and exposure to ICT terms. The testing is carried out during the Beta Phase of The Purposed Game Development Life Cycle (GDLC) process, with the purpose of ensuring that the project is well received by users. This research will be evaluated using the Game User Experience Satisfaction Scale (GUESS) questionnaire developed by Phan, M. H., Keebler, J. R., and Chaparro, B. S. (2016). User demographics, usability/playability, play engrossment/enjoyment visual and aural aesthetics, personal pleasure, and player feedback are all being evaluated.

As indicated below, the usability testing plan for the entire project explains the items needed to run the test, such as the test's purpose, time and location, equipment,

participants, and other factors. However, based on the participant's availability, the date and venue may change slightly.

<table>
<tr><td colspan="3">Developer: FARHAN AKMAL BIN YAHYA</td><td colspan="2">Product Tested:
Vocab Defender, A Shooting Typing Game</td></tr>
<tr><td>Test objectives</td><td>Participants</td><td>Equipment</td><td>Test Task and Duration</td><td>Location and Dates</td></tr>
<tr><td>To test the usability of typing game and user satisfaction/experience while respondents play the game</td><td>Involving respondents from age 13 years old and above
And 2 experts in IT and Multimedia</td><td>Vocab Defender can be played only on personal computer (PC) and laptop.</td><td>Duration: Estimate 35 minutes per person
1. Start play the game. Defeat the wave of enemies by typing all words appear on the screen.
2. Answer the survey after finish play the game.</td><td>At respondents respective residences. Spread the survey start from 1st August</td></tr>
<tr><td colspan="5">Procedure:
1. Welcome and Briefing
5 mins
→
2. Test the game
15 mins
→
3. Answer -test Questionnaire/ Survey
15 mins</td></tr>
</table>

Table 6.1: Usability Testing Plan

6.2.1 Test User

The game will be tested by players ranging in age from 13 years old and above which they are students and employees with perspectives ranging from newbie/novice to hardcore gamer. Other than that, two experts in IT and Multimedia also will involve in this game testing. However, due to a Movement Control Order (MCO) for Covid-19 situation currently, testers can play the game on the website for a better comprehend on the game's main concept and objectives. The testers were then asked to complete a series of questionnaires designed by the developer.

6.2.2 Test Environment

The testing was conducted according the availability of the test participants which it was done at a flexible time. Due to a Movement Control Order (MCO) for Covid-19 situation currently, testers can only do the test which are playing the game and answer the questionnaires online at their respective residences. They just can test the game by using personal computer (PC) or laptop only which the game was built only for those platform.

6.2.3 Test Schedule

The testing will be done in two phases: the first phase will be for the target audience/user, and the second phase will be for expert play testers. However, the test procedure remains the same for each phase, and each test participants will be given 30 to 35 minutes to complete the usability test. As a result, the usability test technique is outlined below..

(i) Before testing session begin :

Developer spreads the game’s link and survey’s link to the participants through the media social like Whatsapp and Telegram.

(ii) During testing session :

a) Thank the participant for participation in the usability testing.

b) Developer will briefly explain what they are asked to do:

- 15 to 20 minutes to carry out the game test online.
- 10 to 15 minutes to do the test questionnaire online.

c) Tester start play/test the game.

(iii) After testing session:

a) The survey data will be stored online after testers done the testing and will be received by developer for data analysis

b) Thank them for participating the usability test.

6.3 Test Strategy

Usability testing can be carried out in a variety of ways. During the usability testing, however, only one method will be used: unmoderated remote usability testing. This testing takes place in the absence of a moderator. It gives user testing findings that are quick, dependable, and low-cost, and can be used for data analysis. Participants are asked to complete activities in their own environment, with their own devices, and without the presence of a moderator, resulting in natural product usage. For this project, testers will be sent links to the game and survey via social media platforms such as Whatsapp and Telegram. The test will be conducted at the testers' homes and on their own time. The survey data will be stored online once the testers have finished testing the game and answering the questionnaires, and the developer will have access to it for data analysis.

As a result, the usability testing strategy is to perform the test online and collect survey data from the testers in order to acquire qualitative and quantitative data.

6.4 Test Implementation

The test implementation process began with a game tester registration, in which the developer sought out players who were interested in participating as testers for this project. The game's website link will then be sent to the group of enrolled testers, who will have 48 hours to play it. The testers must report the test completion to get the survey's link and answer a set of questionnaires.

This test is implemented by three phase which are Involvement Registration, 48 Hours Playing Time and Answering Questionnaires:

1) Involvement Registration

- I introduce myself and explain about the game details such as game genre and game platform to the participants through Whatsapp .

2) 48 Hours Playing Time

- The participants from the poll then finalize and will be given the game's website link to play within 48 hours time.

3) Answer Questionnaires

- The tester who finish the testing phase will then report to the developer about the completion and get the survey's link.
- The tester need to answer a set of questionnaires with total of 24 questions

6.4.1 Test Description

The Game User Experience Satisfaction Scale (GUESS) was created by Phan, M. H., Keebler, J. R., and Chaparro, B. S. (2016) is used to evaluate the game. A total of 9 factors in GUESS were considered for evaluating the game in the purposed questionnaire, which are Usability/Playability, Narratives, Play Engrossment, Enjoyment, Creative Freedom, Audio Aesthetics, Personal Gratification, Social Connectivity, and Visual Aesthetics, but six factors were chosen based on their relevance for the game evaluation test. Usability/Playability, Play Engrossment, Enjoyment, Audio Aesthetics, Personal

Gratification and Visual Aesthetics are the factors that have been picked and Game Effectiveness as extra factor to evaluate the game effectiveness in achieving project's objectives. Its purpose is to assess the game's usability and satisfaction. Furthermore, this testing is used to learn more about the product, provide feedback or suggestions for future improvements, and determine how satisfied people are with the product.

The detailed score scale of the questionnaire is shown in Table 6.2. The questionnaire's scoring system is based on the Linkert Scale, which requires testers to answer questions based on the score and its description.

Description	Strongly Disagree	Disagree	Natural	Agree	Strongly Agree
Score	1	2	3	4	5

Table 6.2 Score Scale for Evaluation Test

Hence, there are several expected results to be achieved in this testing to ease the analysis which are:

(iv) How efficient this game in improving player's typing skills and ICT terms exposure?

(v) How well player's will interact with the game?

(vi) How well player undergo the process of hand (typing) and eye coordination?

6.4.2 Test Data

i. Participant's demographic

The first category is project target audience from age 13 years old and above . Some of them have experienced typing game meanwhile the several are not. The second category is two expert play testers, age around 33 years old and above and both of them have long experienced in multimedia.

Table 6.3 depicts the project evaluation testing included demographic elements from the GUESS questionnaire. The things chosen were thought to be relevant for the project's evaluation testing. The demographic component was chosen to see how demographic data can influence respondents' behaviour and gaming experience. The background of the respondents will assist in determining whether the data can be linearly based on the same demographic group.

Name
Gender
Age
Education Level
Is your education/work involving or using typing skills?
Rate how relevant or often you use typing skills in your education/work
Rate your typing skills level
Which type of video game player is you?
Usual gaming device used?
What the average hours you spent playing game per week?
Types of game main genre that you always play
Have you ever played a typing game?

Table 6.3 Demographic Factors

ii. Data retrieved during test evaluation

Data retrieved from respondents by using a set of GUESS questionnaires which only six factors and one extra self-built factor were chosen and modified based on their relevance for the game evaluation test. Below is the list of factors in GUESS questionnaires that had been used in this project test evaluation (several factors are combined) and the full questionnaires of project's survey can be refer in Appendix 1.

Factors in GUESS questionnaires :

1. Usability/Playability

2. Play Engrossment/ Enjoyment

3. Visual and Audio Aesthetics

4. Personal Gratification

5. Game Effectiveness

6.5 Test and Result Analysis

The goal of the survey questionnaire is to collect data from respondents in seven areas: user demographics, usability/playtesting, play engrossment/enjoyment, visual and audio aesthetic, personal gratification, game effectiveness and player's feedback/suggestions. The purpose of using a questionnaire was to substitute a face-to-face interview which cannot be done in Covid-19 current situation and to corroborate the findings of the observations. The game test and survey will take less than 35 minutes to complete by the respondents.

Section A - Respondents' Background Details

1. Demographics

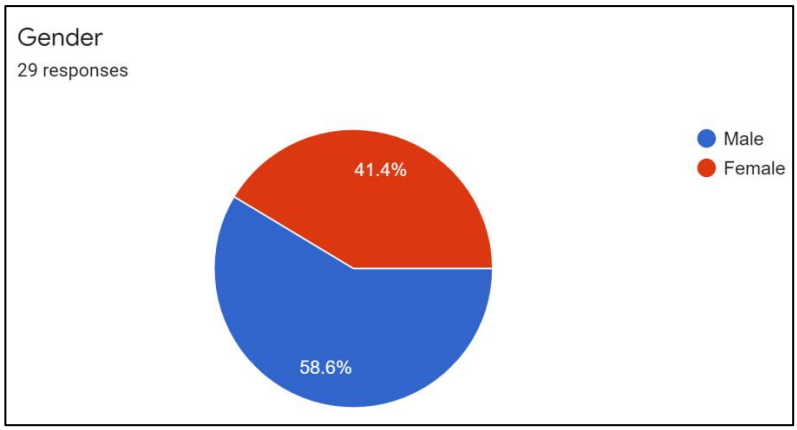

Figure 6.1 Gender Data Pie Chart

Figure 6.1 shows, there are 17 male respondents (58.6%) and 12 female respondents (41.4%) out of total 29 respondents who participate in the testing. As for the majority gender for game tester is male (58.6%).

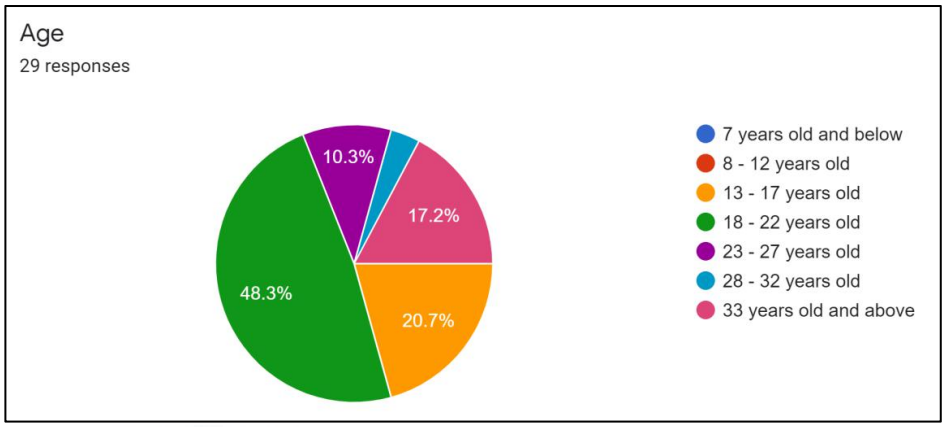

Figure 6.2 Age Data Pie Chart

Figure 6.2 shows, out of 29 respondents participate there are 6 respondent (20.7%) from age 13 - 17 years old which are considered as secondary school students, 14 respondents (48.3%) from age 18 - 22 years old and 3 respondents (10.3%) from age 23 - 27 years old which can be considered as university/college students, 1 respondent (3.4%) from age 28 - 32 years old and 5 respondents (17.2%) from age 33 years old and above will be consider as workers. As for the majority age that has been tester is between 18 - 22 years (48.3%).

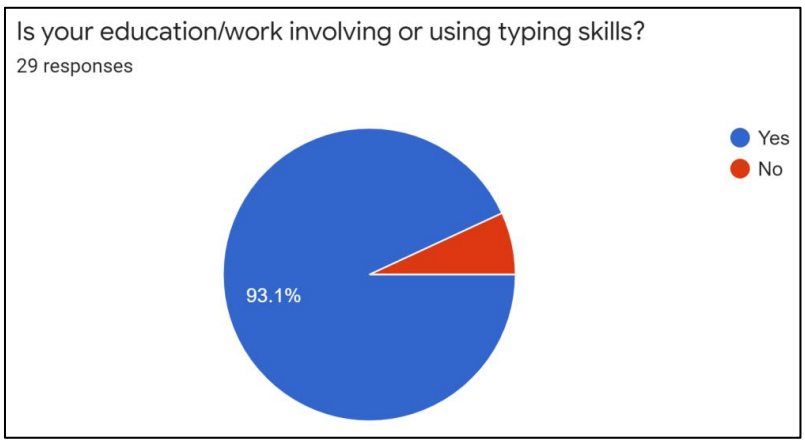

Figure 6.3 Typing Skills Usage Pie Chart

In Figure 6.3, almost all respondents which are 27 respondents (93.1%) stated that typing skills is being used in their education or work while the remaining, 2 respondents (6.9%) stated not used typing skills.

Questionnaire	Mean	Standard Deviation
how relevant or often you use typing skills in your education/work	4.103	0.8596
Rate your typing skills level	6.862	1.6196

Table 6.4 : Typing Skills Usage and Level Mean and Standard Deviation

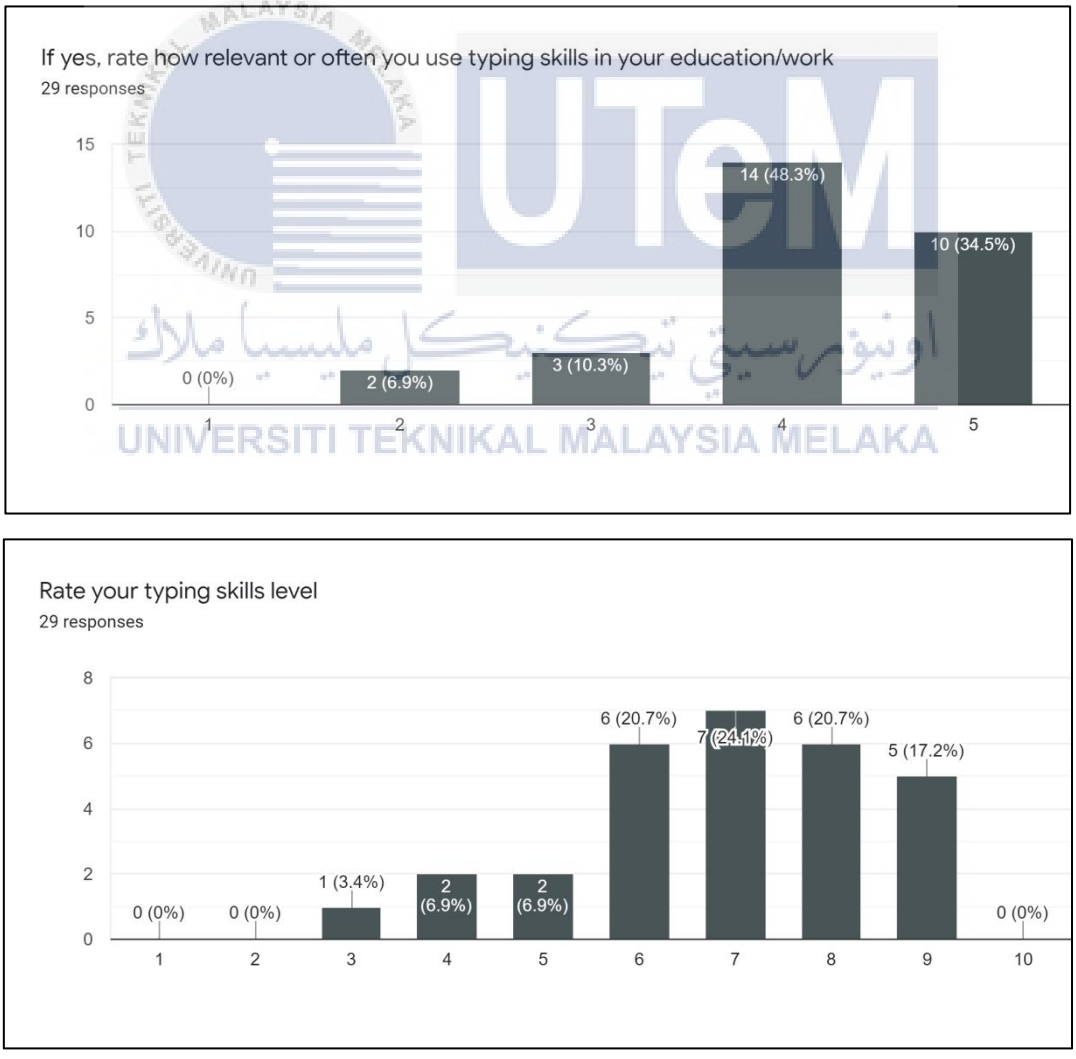

Figure 6.4 : Typing Skills Usage and Level Bar Graph

In Table 6.4 and Figure 6.4, we can see that 24 respondents chose 4 (often), 48.3% and 5 (very often), 34.5% from total respondents and the mean of typing skills usage is 4 which we can conclude that typing skills is used oftenly by most respondents. Average typing skills level is 6 and 7 respondents (24.1%), the highest number of respondents had rate their typing skills as 7 which means most of respondents have above moderate of typing skills.

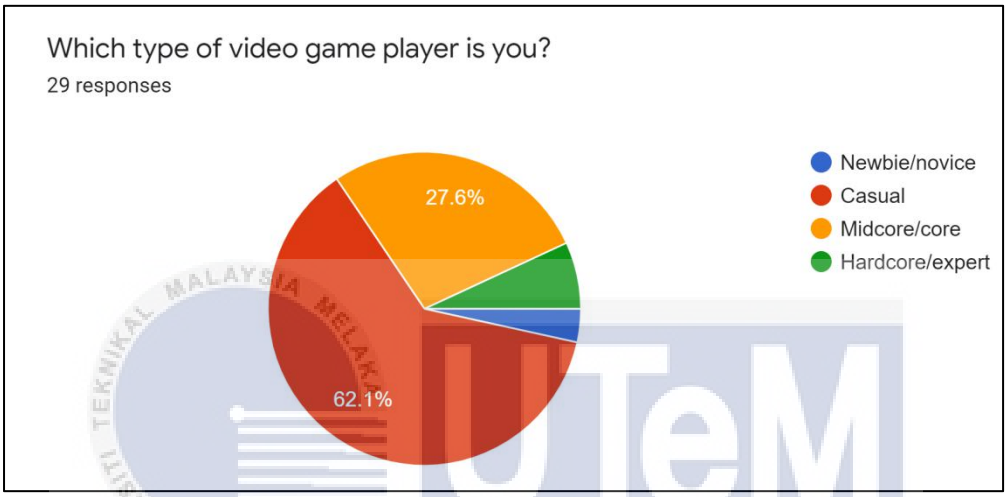

Figure 6.5 : Type of Video Game Player Pie Chart

Figure 6.7 shows that from total 29 respondents, most respondents are Casual Player (18, 62.1%), followed by Modecore/core (8, 27.6%), and Hardcore/expert (2, 6.9%) while only 1 respondent (3,4%) who is Newbie/novice player.

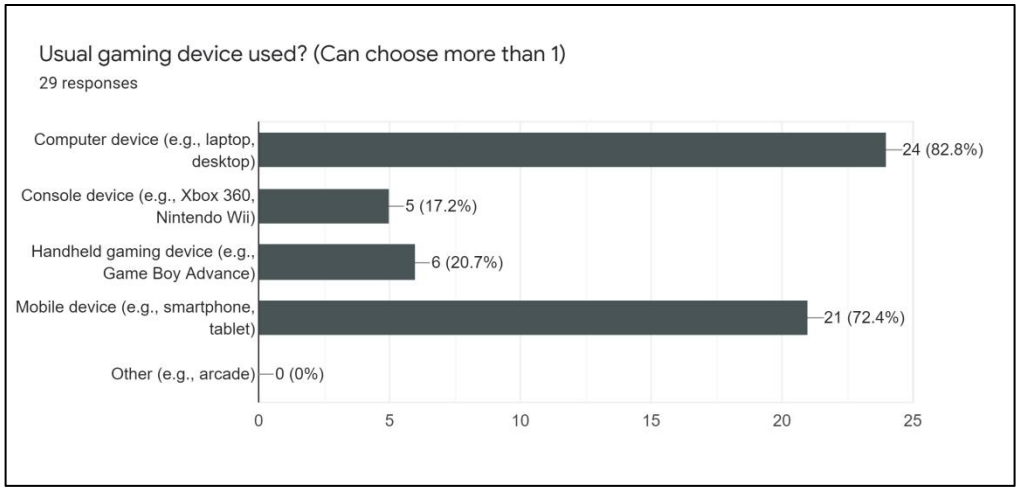

Figure 6.6 : Usual Gaming Device Used Bar Graph

Figure 6.6 shows that most respondents chose computer (24, 82.8%) as usual device for playing game that they used, followed by mobile device (21, 72.4%) and handled gaming device (6, 20.7%) while the least respondents chose console (5, 17.2%).

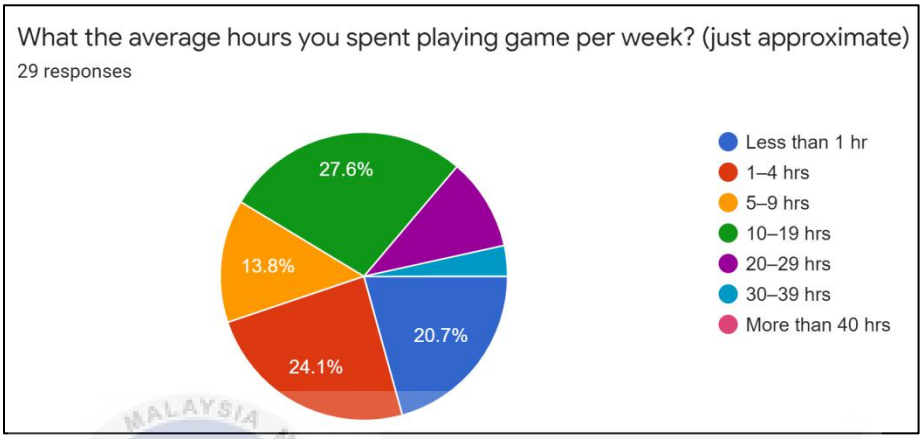

Figure 6.7 : Average Hours Spent Playing Game per Week Pie Chart

Figure 6.7 shows that from total 29 respondents, most respondents spent 10 - 19 hours (8, 27.6%) per week to play games, followed by 1 - 4 hours (7, 24.1%) per week, less than 1 hour (6, 20.7%) per week, 5 - 9 hours (4, 13.8%) per week and 20 - 29 hours (3, 10.3%) per week while the least respondents play games 30 - 39 hours (1, 3.4%) per week.

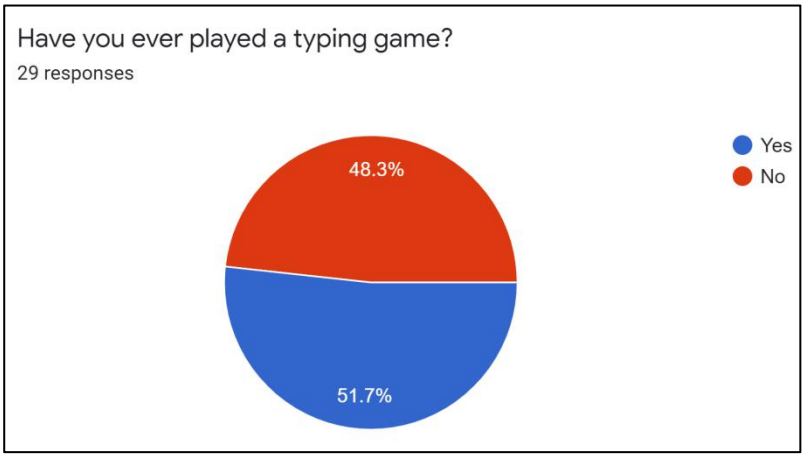

Figure 6.8 : Ever Played Typing Game Pie Chart

Figure 6.8 shows that from total 29 respondents, 15 respondents had played typing game while the remaining 14 respondents have never play typing game.

Section B - GUESS Factors

2. Usability/Playability

i. Respondent (Target User)

Questionnaire	Mean	Standard Deviation
I think it is easy to learn how to play the game.	**4.38**	**0.82**
I find the controls of the game to be straightforward.	**4.24**	**0.95**
I always know how to achieve my goals/objectives in the game.	**4.28**	**0.84**
I find the game's interface to be easy to navigate	**4.03**	**0.94**
I do not need to go through a lengthy tutorial or read a manual to play the game	**4.21**	**1.08**
I find the game's menus to be user friendly.	**3.97**	**0.94**
I feel the game trains me well in all of the controls.	**4.14**	**0.88**
I always know my next goal when I finish an event in the game.	**4.00**	**0.89**
I feel the game provides me the necessary information to accomplish a goal within the game	**4.14**	**0.91**
I think the information provided in the game (e.g., onscreen messages, help) is clear	**4.21**	**0.90**
I feel very confident while playing the game	**4.21**	**0.94**

Table 6.5 : Usability/Playability Questionnaires Mean and Standard Deviation for Target User

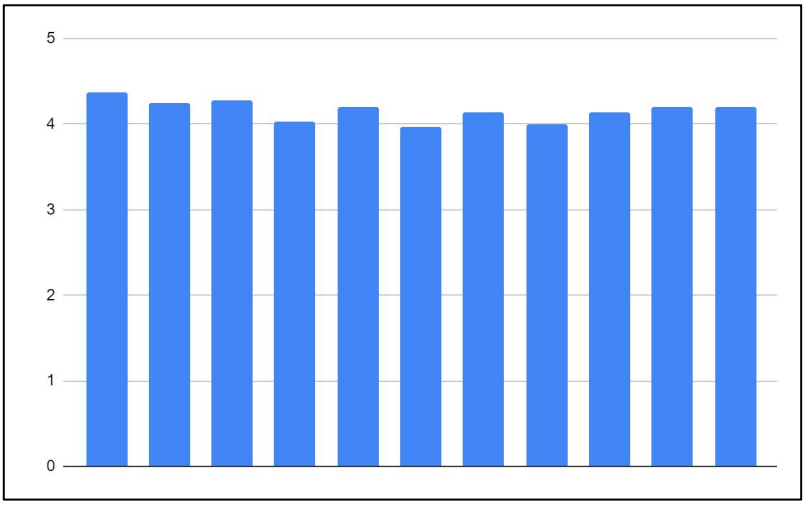

Figure 6.9 : Mean for each Usability/Playability Questionnaires for Target User

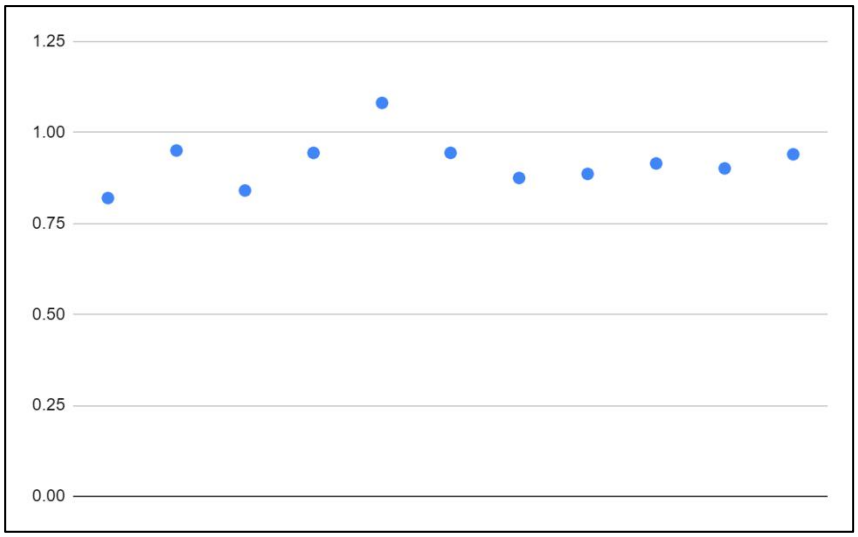

Figure 6.10 : Standard Deviation for each Usability/Playability Questionnaires for Target User

ii. Respondent (Expert)

Questionnaire	Mean	Standard Deviation
I think it is easy to learn how to play the game.	**4.50**	**0.71**
I find the controls of the game to be straightforward.	**4.50**	**0.71**
I always know how to achieve my goals/objectives in the game.	**4.50**	**0.71**
I find the game's interface to be easy to navigate	**3.50**	**0.71**
I do not need to go through a lengthy tutorial or read a manual to play the game	**4.50**	**0.71**
I find the game's menus to be user friendly.	**3.50**	**0.71**
I feel the game trains me well in all of the controls.	**3.50**	**0.71**
I always know my next goal when I finish an event in the game.	**3.50**	**0.71**
I feel the game provides me the necessary information to accomplish a goal within the game	**3.50**	**0.71**
I think the information provided in the game (e.g., onscreen messages, help) is clear	**3.50**	**0.71**
I feel very confident while playing the game	**3.50**	**0.71**

Table 6.6 : Usability/Playability Questionnaires Mean and Standard Deviation for Expert

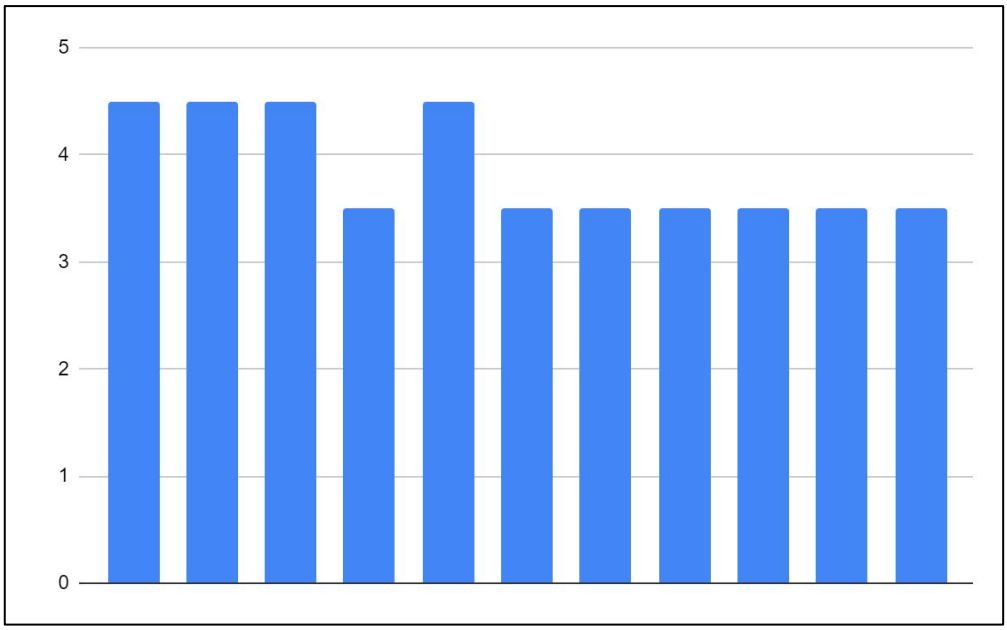

Figure 6.11 : Mean for each Usability/Playability Questionnaires for Expert

There are 11 evaluations based on the mean graph in Figure 6.9 which the questionnaires for Usability/Playability factor are shown in Table 6.5. The mean for most evaluations is between range 4 and above, indicating that the majority of respondents agree with the game's usability except the mean of one questionnaire which is 'I find the game's menus to be user friendly' is 3. The user interface for the game is at moderate level which need to be improved. Based on the findings, we can infer that the system's design and usability are as they should be and yet, it appears to be a little too complex for certain people, particularly those unfamiliar with this genre.

Differ for expert testing, the mean for most evaluation questions are 3 and above based on Table 6.6. This shows the experts less agree or natural with the user interface design and presentation of information in the game.

3. Play Engrossment/Enjoyment

i. Respondent (Target User)

Questionnaire	Mean	Standard Deviation
I think the game is fun.	**4.31**	**0.93**
I enjoy playing the game.	**4.41**	**0.82**
I never feel bored while playing the game.	**3.76**	**0.99**
I think the gameplay and mechanics are interesting	**4.03**	**0.94**
I am likely to recommend this game to others.	**4.103**	**1.05**
If given the chance, I want to play this game again.	**4.00**	**0.85**
I feel detached from the outside world while playing the game	**3.3**	**1.04**
I cannot tell that I am getting tired while playing the game	**3.67**	**1.00**
I temporarily forget about my everyday worries while playing the game.	**3.34**	**1.14**
I tend to spend more time playing the game than I have planned.	**3.45**	**1.18**
I can block out most other distractions when playing the game	**3.41**	**1.18**
Whenever I stopped playing the game I cannot wait to start playing it again.	**3.31**	**1.28**

Table 6.7 : Play Engrossment/Enjoyment Questionnaires Mean and Standard Deviation for Target User

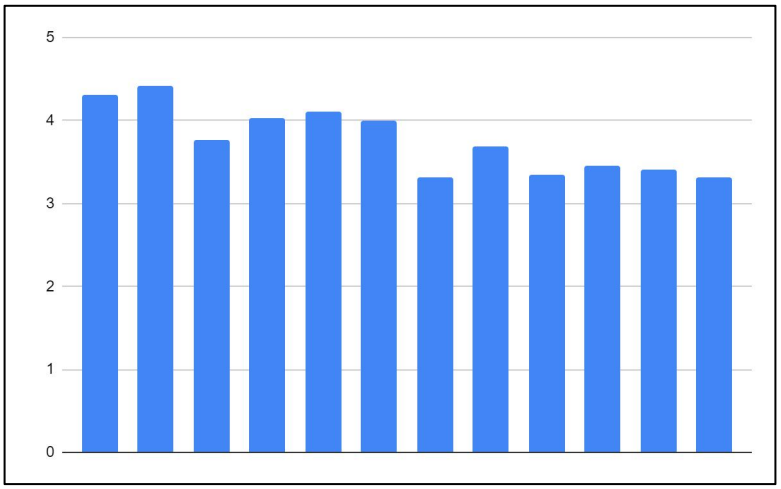

Figure 6.12 : Mean for each Play Engrossment/Enjoyment Questionnaires for Target User

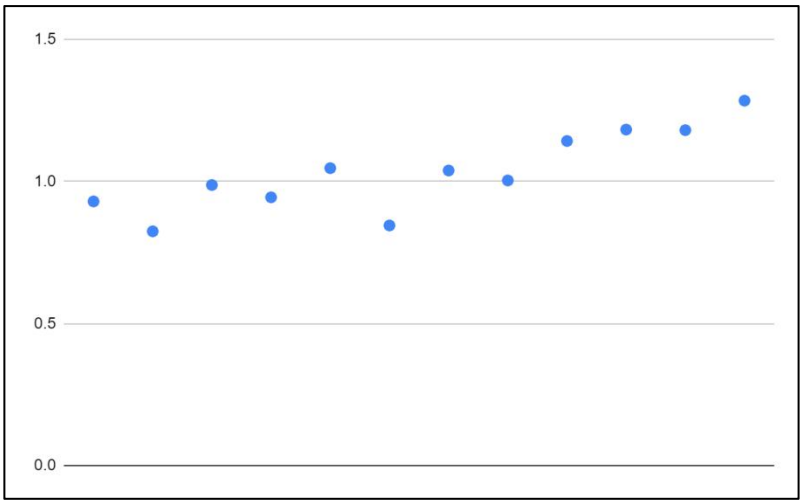

Figure 6.13 : Standard Deviation for each Play Engrossment/Enjoyment Questionnaires for Target User

ii. Respondent (Expert)

Questionnaire	Mean	Standard Deviation
I think the game is fun.	**3.50**	**0.71**
I enjoy playing the game.	**3.50**	**0.71**
I never feel bored while playing the game.	**3.50**	**0.71**
I think the gameplay and mechanics are interesting	**3.50**	**0.71**
I am likely to recommend this game to others.	**3.50**	**0.71**
If given the chance, I want to play this game again.	**3.50**	**0.71**
I feel detached from the outside world while playing the game	**3.50**	**0.71**
I cannot tell that I am getting tired while playing the game	**3.50**	**0.71**
I temporarily forget about my everyday worries while playing the game.	**3.50**	**0.71**
I tend to spend more time playing the game than I have planned.	**3.50**	**0.71**
I can block out most other distractions when playing the game	**3.50**	**0.71**
Whenever I stopped playing the game I cannot wait to start playing it again.	**3.50**	**0.71**

Table 6.8 : Play Engrossment/Enjoyment Questionnaires Mean and Standard Deviation for Expert

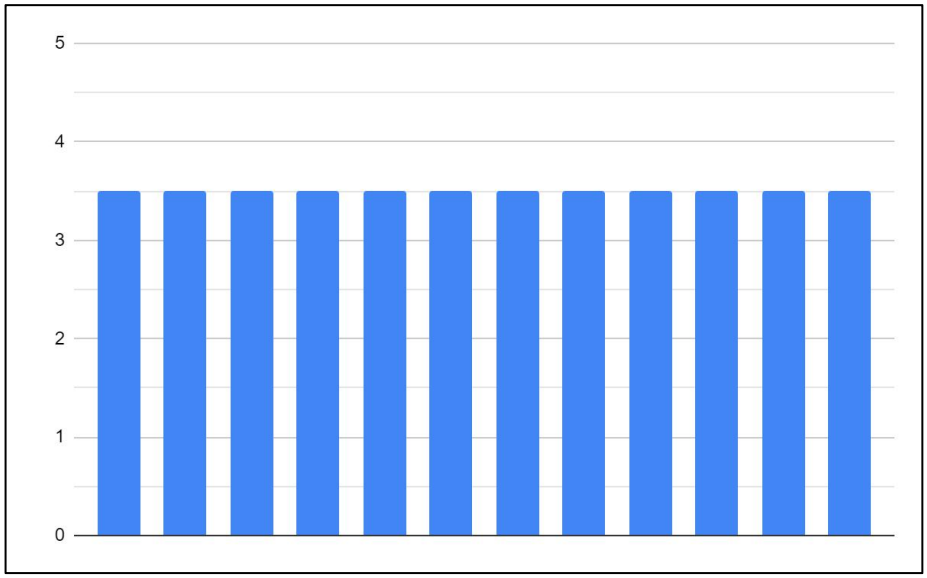

Figure 6.14 : Mean for each Play Engrossment/Enjoyment Questionnaires for Expert

According to the mean graph in Figure 6.12, there are 12 evaluations which are shown in Table 6.7. As can be seen, the first six evaluations have mean scores in the range of 4 and above, indicating that the majority of respondents agree that the game give enjoyment and fun to play but one evaluation which is 'I never feel bored while playing the game' had a mean 3 which mean the level of boredom is in the middle. Maybe a part of the respondents will easily feel bored. For the remaining evaluations, all the mean are 3 and above which means the level of Play Engrossment or immersion level is moderate. The game is not to immersive which make some respondents to get attach with the game for a long period, they can be stopped. As a consequence of the findings, we can conclude that the majority of players enjoy the game due to the function of game mechanics which is typing mechanisms, but not for a long period. Despite that, the game still can be a practice tool for sort of practice time that be done frequently.

Same goes for expert testing, the mean for all evaluation questions are 3 and above based on Table 6.8. This shows the experts have moderate immerse to the game. It can be that it is influenced by the test result from Usability factor which mention that the game's menu and user interface is less user-friendly.

4. Visual and Audio Aesthetics

i. Respondent (Target User)

Questionnaire	Mean	Standard Deviation
I enjoy the game's graphics.	**3.97**	**1.02**
I think the game theme and design is interesting	**4.07**	**1.03**
I think the graphics of the game fit the mood or style of the game.	**4.03**	**1.05**
I think the game is visually appealing.	**3.93**	**1.07**
I enjoy the sound effects in the game.	**4.03**	**1.12**
I enjoy the music in the game.	**4.28**	**0.92**
I feel the game's audio (e.g., sound effects, music) enhances my gaming experience	**4.00**	**1.10**
I think the game's audio fits the mood or style of the game	**4.24**	**0.95**

Table 6.9 : Visual and Audio Aesthetics Questionnaires Mean and Standard Deviation for Target User

Figure 6.15 : Mean for each Visual and Audio Aesthetics Questionnaires for Target User

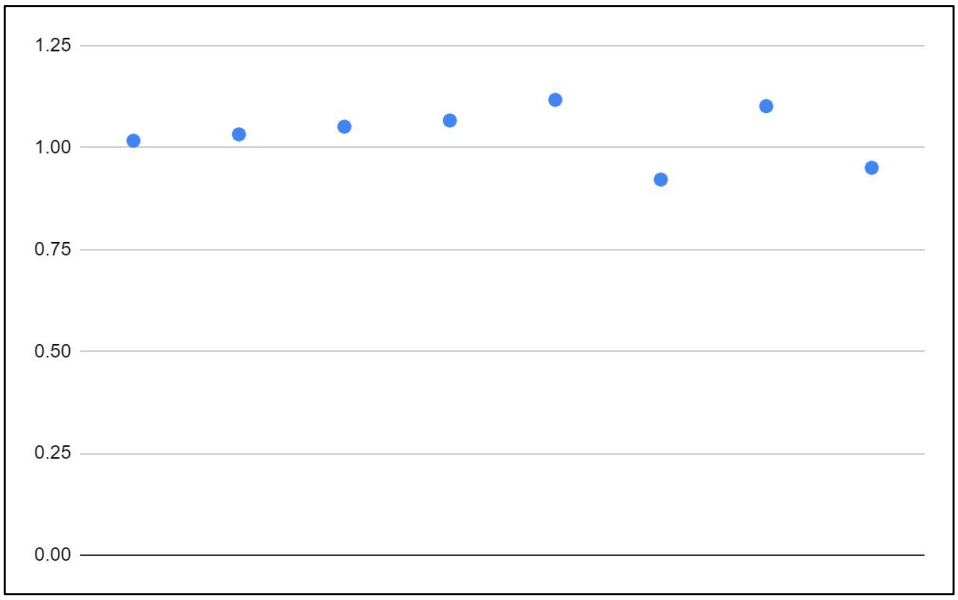

Figure 6.16 : Standard Deviation for each Visual and Audio Aesthetics Questionnaires for Target User

ii. Respondent (Expert)

Questionnaire	Mean	Standard Deviation
I enjoy the game's graphics.	**3.00**	**1.41**
I think the game theme and design is interesting	**3.00**	**1.41**
I think the graphics of the game fit the mood or style of the game.	**3.00**	**1.41**
I think the game is visually appealing.	**3.00**	**1.41**
I enjoy the sound effects in the game.	**3.00**	**1.41**
I enjoy the music in the game.	**3.00**	**1.41**
I feel the game's audio (e.g., sound effects, music) enhances my gaming experience	**3.00**	**1.41**
I think the game's audio fits the mood or style of the game	**3.00**	**1.41**

Table 6.10 : Visual and Audio Aesthetics Questionnaires Mean and Standard Deviation for Target User

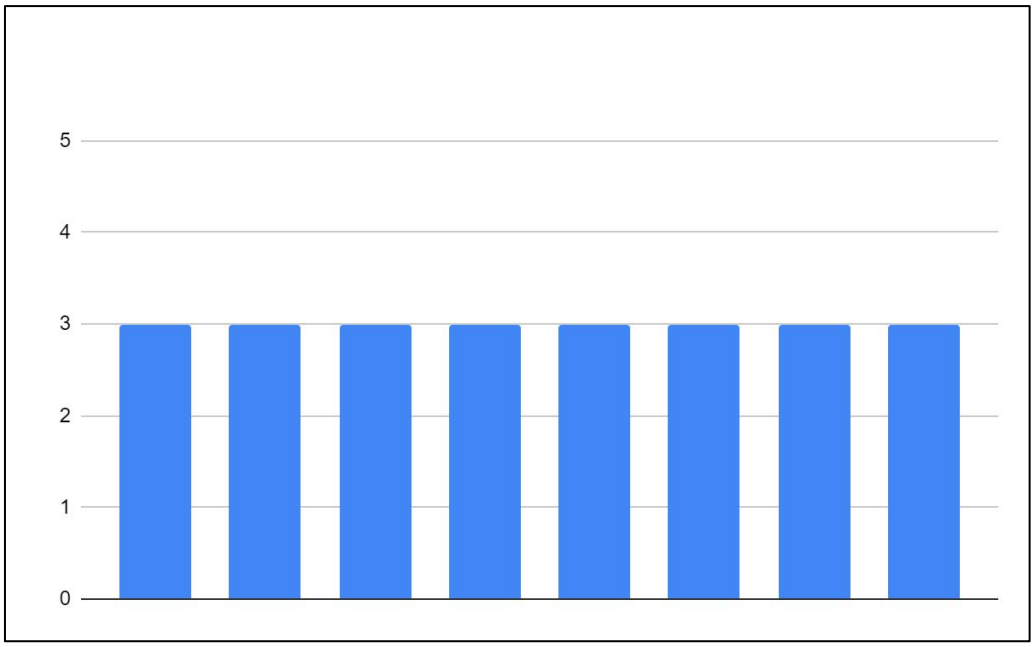

Figure 6.17 : Mean for each Visual and Audio Aesthetics Questionnaires for Expert

Based on Table 6.9 and the mean graph in Figure 6.15, most respondents agree with the game theme and design is interesting and how well the graphics merge and suit the mood and style in the game, that can be shown by the mean for both evaluations are 4 and above but for visual appealing and the enjoyment for game's graphics are moderate which the mean for both are 3 and above. The majority of respondents agree with the assessment, yet we can infer that the game's graphic and visual presentation still has some flaws that prevent users from being completely satisfied. As for audio aesthetics, all the mean for the audio evaluations are 4 and above. It can be said that respondent quite satisfied with audio usage in the game and it also can be improved for a better enjoyment and comfort play by players.

Differ for expert testing, the mean for all evaluation questions are 3 and above based on Table 6.10. It can be that it is influenced by the test result from Usability factor which mention that the game's menu and user interface is less user-friendly. The experts' play are little affected by the game's visual and audio.

5. Personal Gratification

i. Respondent (Target User)

Questionnaire	Mean	Standard Deviation
I am in suspense about whether I will succeed in the game	**3.83**	**1.07**
I feel successful when I overcome the obstacles in the game.	**4.20**	**0.86**
I want to do well as possible during the game.	**4.38**	**0.78**
I am very focused on my own performance while playing the game.	**4.34**	**0.89**
I feel the game constantly motivates me to proceed further to the next stage or level	**4.17**	**1.04**
I find my skills gradually improve through the course of overcoming the challenges in the game	**4.24**	**0.91**

Table 6.11 : Personal Gratification Questionnaires Mean and Standard Deviation for Target User

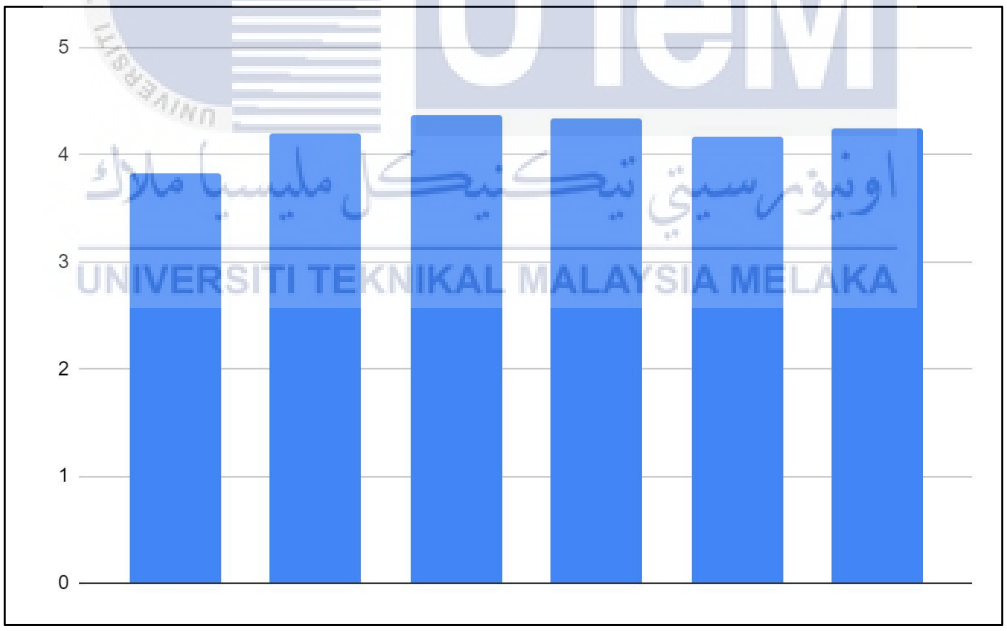

Figure 6.18 : Mean for each Personal Gratification Questionnaires for Target User

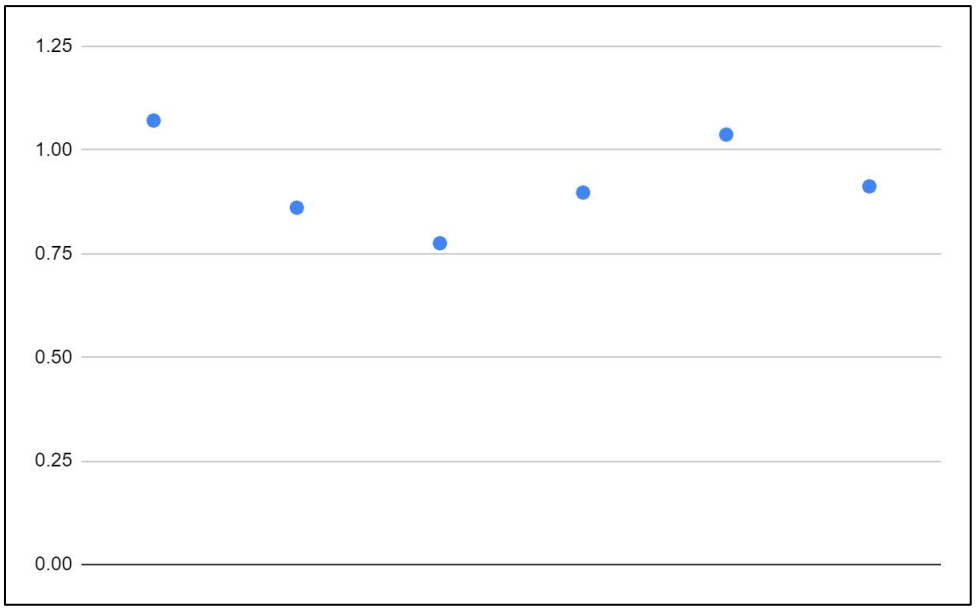

Figure 6.19 : Standard Deviation for each Personal Gratification Questionnaires for Target User

ii. Respondent (Expert)

Questionnaire	Mean	Standard Deviation
I am in suspense about whether I will succeed in the game	**4.50**	**1.71**
I feel successful when I overcome the obstacles in the game.	**4.50**	**1.71**
I want to do well as possible during the game.	**4.50**	**1.71**
I am very focused on my own performance while playing the game.	**4.50**	**1.71**
I feel the game constantly motivates me to proceed further to the next stage or level	**4.50**	**1.71**
I find my skills gradually improve through the course of overcoming the challenges in the game	**4.00**	**0.00**

Table 6.12 : Personal Gratification Questionnaires Mean and Standard Deviation for Expert

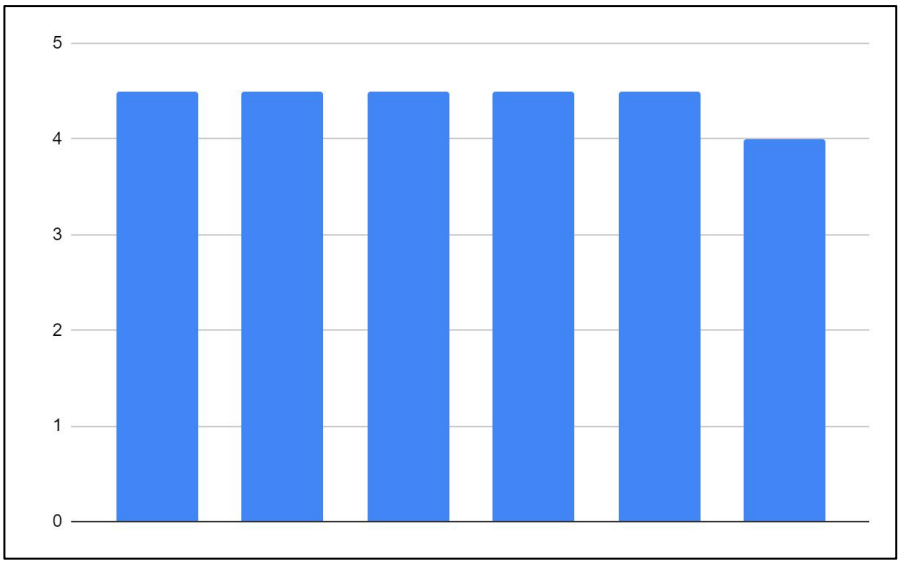

Figure 6.20 : Mean for each Personal Gratification Questionnaires for Expert

Based on Table 6.11 and the mean graph in Figure 6.18 , most respondents satisfied while play the game is shown by the mean for all evaluations are 4 and above except for 'I am in suspense about whether I will succeed in the game' questionnaire which its mean is 3 and above. The majority of respondents complacent while play the game, although the suspense while play the game is moderate which not too hard to give more challenge to player. The game's difficulty need to be enhanced and be various so that suitable for all player type.

Differ for expert testing in Table 6.12, the mean for all evaluation questions are 4 and above because the experts not feel to suspense because the experts' skills is higher than normal respondents.

6. Game Effectiveness

i. Respondent (Target User)

Questionnaire	Mean	Standard Deviation
I can measure and know my current typing skill level and its improvement by playing this game	**4.17**	**1.10**
I am sure that I can improve my typing skill if I keep playing this game	**4.38**	**0.86**
This game makes me motivated to keep improving my typing skills	**4.31**	**0.81**
The harder the game's difficulty/challenge, the more I want to do better in improving my typing skills	**4.34**	**0.72**
I am exposed and can identify ICT terms in the game	**4.31**	**0.93**
I am sure that I can improve my ICT terms usage by playing this game	**4.21**	**0.90**
While I keep playing this game, I can detect and memorize some ICT terms used in the game	**4.17**	**0.80**
The gameplay and scoreboard system is interesting and helpful which make me keep playing to improve my typing skills and exposure to the ICT terms	**4.38**	**0.68**
I think this game suitable for school and university students	**4.41**	**0.87**
I think this game suitable for the workers	**4.17**	**0.97**
I think this game suitable for all ages who want improve their typing skills	**4.45**	**0.87**
I think the ICT terms used is suitable with game theme	**4.31**	**0.97**

Table 6.13 : Game Effectiveness Aesthetics Questionnaires Mean and Standard Deviation for Target User

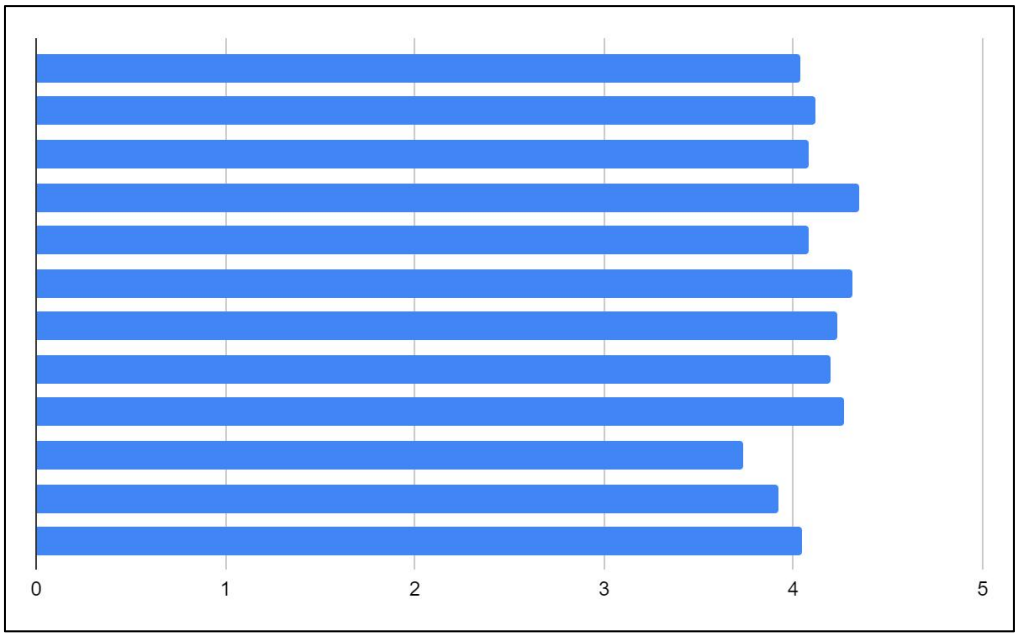

Figure 6.21 : Mean for each Game Effectiveness Questionnaires for Target User

Figure 6.22 : Standard Deviation for each Game Effectiveness Questionnaires for Target User

ii. Respondent (Expert)

Questionnaire	Mean	Standard Deviation
I can measure and know my current typing skill level and its improvement by playing this game	**4.00**	**0.00**
I am sure that I can improve my typing skill if I keep playing this game	**4.50**	**1.71**
This game makes me motivated to keep improving my typing skills	**4.00**	**0.00**
The harder the game's difficulty/challenge, the more I want to do better in improving my typing skills	**4.00**	**0.00**
I am exposed and can identify ICT terms in the game	**4.50**	**1.71**
I am sure that I can improve my ICT terms usage by playing this game	**4.50**	**1.71**
While I keep playing this game, I can detect and memorize some ICT terms used in the game	**4.50**	**1.71**
The gameplay and scoreboard system is interesting and helpful which make me keep playing to improve my typing skills and exposure to the ICT terms	**4.50**	**1.71**
I think this game suitable for school and university students	**4.50**	**1.71**
I think this game suitable for the workers	**4.00**	**0.00**
I think this game suitable for all ages who want improve their typing skills	**3.50**	**1.71**
I think the ICT terms used is suitable with game theme	**3.50**	**1.71**

Table 6.14 : Game Effectiveness Aesthetics Questionnaires Mean and Standard Deviation for Expert

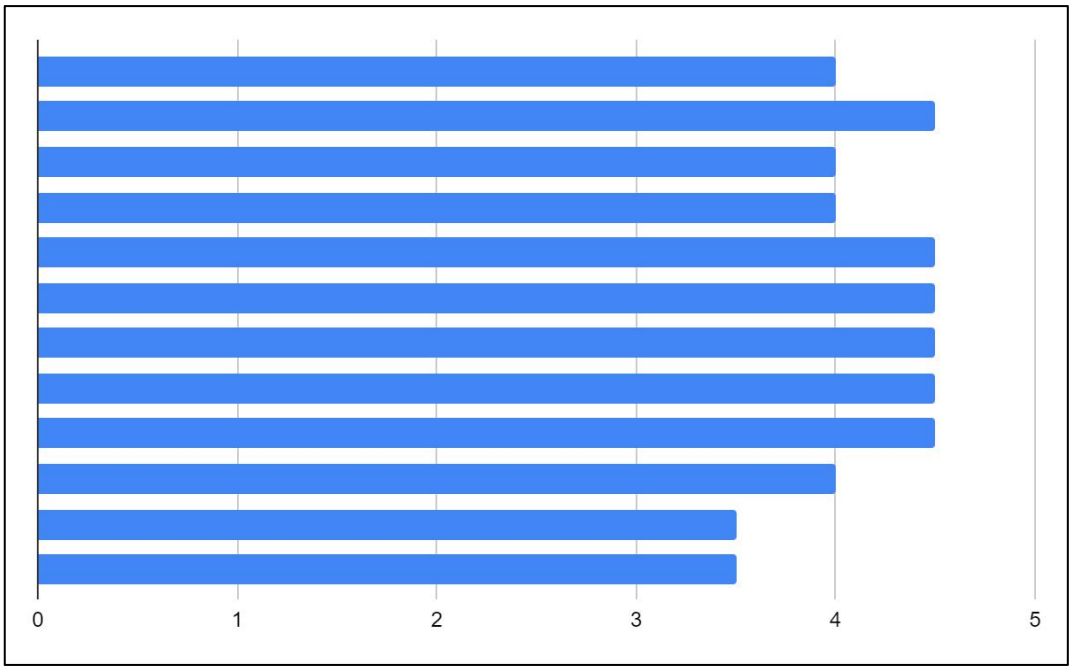

Figure 6.23 : Mean for each Game Effectiveness Questionnaires for Expert

Based on the mean graph in Figure 6.21, most respondents are agree that the game is suitable for typing skills improvement along with suitable amount practice time. This can be seen by the mean for all evaluations are 4 and above in Table 6.9. The majority of respondents complacent and agree that Vocab Defender typing game can be a practice tool in enhancing players' typing skills but with a correct amount of practice time and consistent in practice, yet there are a lot of improvements need to be done in the game for a better play experience. Most respondents also agree that the game is suitable for all ages especially students who nowadays done many works involving typing skills using computer or laptop and other devices and indirectly had achieve the project's objectives.

Same goes for expert testing, the mean for all evaluation questions are 4 and above except for 2 questions which had the mean 3 and above based on Table 6.14. This shows the experts agree that the game can help in improving players' typing skills but it most suitable for students and workers. It can be that the mean for the last two questions are 3 and above because it is influenced by the test result from Usability factor and Visual and Audio Aesthetics factor which mention that the game's menu and user interface is less user-friendly

Overall Test Result based on the Factors in GUESS

I. Respondent (Target User)

GUESS Factor	Mean	Standard Deviation
Usability/Playability	**4.16**	**0.91**
Play Engrossment / Enjoyment	**3.76**	**1.03**
Visual and Audio Aesthetics	**4.07**	**1.03**
Personal Gratification	**4.19**	**0.93**
Game Effectiveness	**4.30**	**0.87**

Table 6.15 : Overall Test Result for Target User

II. Respondent (Expert)

GUESS Factor	Mean	Standard Deviation
Usability/Playability	**3.86**	**0.71**
Play Engrossment / Enjoyment	**3.50**	**0.71**
Visual and Audio Aesthetics	**3.00**	**1.41**
Personal Gratification	**4.42**	**0.59**
Game Effectiveness	**4.17**	**0.47**

Table 6.16 : Overall Test Result for Expert

Based on Table 6.15, overall test result for 5 factors from target user and from expert show that respondents and experts agree that game is suitable as a practise tool and helps in improving players' typing skills and ICT terms exposure which directly state that the project's objectives are achieved but the game needs improvements from time to time for a better play experience and effectiveness.

7. **Respondents' and Experts' Feedback/Opinion and Suggestions**

From the total 29 respondents, most respondents give overall feedback which said Vocab Defender typing game has a good basic mechanics and suitable for who love typing game and want to improve their typing skills. Several of them were also giving suggestion for the further improvement. For the experts, they give a feedback in detail on the game's strengths and weaknesses and several suggestions to make the game be more acceptance and enjoy to be played beside gain benefit to improve typing skills. Full result on feedback and suggestions can be referred at Appendix B.

6.6 Conclusion

As a conclusion, this chapter briefly describes the testing and responses from respondents who participated in testing the game before filling out a survey. A total of 24 questions with a few divisions of questionnaires for each factors were asked, with 29 people responding. The majority of respondents firmly believe that the game is positive in all aspects, according to the data. According to the respondents' suggestions, the game may be improved by increasing the game's user interface, which is making it more organize and various the game theme and color scheme, as well as adding more mechanics to make the game more interesting. The observation of strengths and flaws, as well as suggestions for improvement and contribution, will be covered in the following chapter.

CHAPTER 7: PROJECT CONCLUSION

7.1 Observation on Weaknesses and Strengths

The strengths in my project are the gameplay and game mechanics. The game mechanics of typing game which is typing mechanisms is very useful for typing skills enhancement if practice consistently and the gameplay of shooting the enemies is for the player to be more enjoying the game while perform the main objectives which is typing all the words appear on the screen. Furthermore, the usage of audio like shooting sound and typing sound effect and also various background music that players can choose is also one of my game's strength. The variety of background music and sound effects can make the game more immersive and rich in contents. Player can choose the suitable background music which indirectly can build a good mood for the players performs better on the game. The most of respondents are satisfied with this game and enjoy to play it although there are still several parts need to be improved. Below is several respondents' opinions on the advantage of the game.

What are the advantages of the game in general or in improving players' typing skills? Justify your opinion.
The basic mechanic of typing game that you have developed is there and good to go
The graphics and sound really attract player to play the game and try to improve their typing skill.
The game play and game mechanic itself helps the player's to improving their typing skills. In fact, at the end of the game, it shown the player's WPM and accuracy.

As for weaknesses from self observation and user feedback. I can conclude that the game have major and minor flaws. The major weakness in the game are user interface and the color scheme usage. From the evaluation of the survey, the user interfaces in the game are not too user-friendly especially in-game UI which player need a few time to navigate or discover all the functional elements like buttons while play the game. Other than that, a few of respondents give a feedback that they not very attach or not very pleased with the color of game world. There is a respondent said that she miss a lot of things while she just can focusing on typing only and there is no sign or warning if players type a wrong letter. For minor flaw is the performance of the game is a bit lag for sometimes and there is no a detail tutorial or how to play for a better understanding. So by adding more options to the player can solve this problem. Below is several respondents' opinions on the weaknesses of the game.

What are the weaknesses of the game? Justify your opinion.
The game do not alert the player visually when they press the wrong button
Confusing for the novice. Should stated clear instruction or game demo.
game UI and UX. somehow it needs improvement

7.2 Propositions of Improvement

With the addition of user feedback and suggestions displayed in Appendix C, there are several propositions that are eligible to be considered as future improvements based on the testing evaluation.

First major improvement is improving the User Interface (UI) design and color scheme of game theme in game world. The world environment UI also should be more variety that can make player feels ‘user-friendly’. The color scheme can be change by player to suit their condition and play style. Other than that, this game can be added more various game mechanics like power ups to make the game more lively and can attract more players’ attention to keep playing this game.

Second improvement is the game can be play on various platforms like tablets and smartphone which both of them usually used especially by students for their online PDPR during this pandemic because not everyone can afford a computer or a laptop. Besides players can improve their typing skills using actual keyboard, they also can practice typing on virtual keyboard. The difficulty of the game also need to modify to give a various option to the players.

Other than that are the minor improvement like the game should provide better information on how to play and add a suitable sign for a few functions in game like spawn an alert if players type a wrong letter. Moreover, this game can be improved with make or add more variety of game mechanisms like add power up or unlock new ability to make the game more attractive and immersive. This variety of game mechanisms is critical in order to provide a free atmosphere in which players can do various things in order to enjoy and improving gameplay experiences while also gain benefits from the game.

7.3 Project Contribution

The goal of this project is actually intended to produce a game using a technology or ICT theme and content as material, as well as typing mechanics as a tool for improving typing skills. Furthermore, this product will unintentionally demonstrate that it may be played and utilized as a typing practise tool by people of all ages, from students to workers, who want to improve their typing skills and promote or expose rapid technological growth in ICT, particularly to young generation. The goal of this project is to see how successful a typing game is at improving players' typing abilities as it is one of important skills which very useful in

ease our works that involve computer and the words used in the game are exposing them to ICT terms.

7.4 Conclusion

In conclusion, the major goals of Vocab Defender, a shooting typing game, have been achieved, allowing players to develop their typing skills and gain exposure to ICT terms and concepts through appropriate and sufficient practise hours. The product also has a lot of potential and can be enhanced and polished to give a better play experience and effectiveness to the players and increase its marketability in the game industry in Malaysia and even beyond.

REFERENCES

Fodor, Szabina & Varga, Márton. (2020). Using Gamification to Improve Students' Typing Skills. 10.1007/978-3-030-63464-3_19.

Ikhsananto, A., & Sutirman, S. (2018). The Effectiveness of Rapid Typing Software and Module for Improving 10-Finger Typing Skills. Dinamika Pendidikan, 13(2), 228-237. doi:https://doi.org/10.15294/dp.v13i2.15540

Nor Hazwani Binti Abdul Rahman. (2015). Kid's Online Typing Game (Cabaran Keyboard).http://umpir.ump.edu.my/id/eprint/12101/1/FSKKP%20-%20NOR%20HAZWANI%20ABDUL%20RAHMAN%20(CD9172).pdf

Lin CH., Liu E.ZF. (2009) Do Improve Typing Skill But No Significant Difference between Drill-Based and Game-Based Typing Software. In: Chang M., Kuo R., Kinshuk, Chen GD., Hirose M. (eds) Learning by Playing. Game-based Education System Design and Development. Edutainment 2009. Lecture Notes in Computer Science, vol 5670. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-03364-3_19

Nyoman Sumerti, Budi Susetyo. (2019). MaxType Media To Improve Typing Skills Use 10 Fingers For Deaf Childrens.

Hosch, W. L. (1978). *Electronic shooter game. 1979.*

Ramadan, R., & Widyani, Y. (2013). Game development life cycle guidelines. *2013 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, *September 2013*, 95–100. https://doi.org/10.1109/ICACSIS.2013.6761558

Mohammed, Phaedra & Mohan, Permanand & Augustine, St & Trinidad, Tobago. (2010). Combining Digital Games with Culture: A Novel Approach towards Boosting

Student Interest and Skill Development in Computer Science Programming. Mobile, Hybrid, and On-line Learning, International Conference on. 10.1109/eLmL.2010.20.

Phan, Mikki & Keebler, Joseph & Chaparro, Barbara. (2016). The Development and Validation of the Game User Experience Satisfaction Scale (GUESS). Human Factors: The Journal of the Human Factors and Ergonomics Society. 58. 10.1177/0018720816669646. https://www.researchgate.net/publication/308343588_The_Development_and_Validation_of_the_Game_User_Experience_Satisfaction_Scale_GUESS

Appendix A

Pengesahan Pakar Bidang

Adalah disahkan bahawa kesemua kandungan soalan ujian dan komponen projek (permainan video) yang telah dibina oleh saudara Farhan Akmal bin Yahya (B031810377) dari Fakulti Teknologi Maklumat Komunikasi, Universiti Teknikal Malaysia Melaka, sesuai bagi golongan pelajar dan pekerja yang banyak menggunakan kemahiran menaip.

Sekian, terima kasih.

Ulasan Am:

Rujuk google form

Nama : DR MOHAMAD LUTFI DOLHALIT

Jawatan : PENSYARAH KANAN

Tempat Bertugas : UNIVERSITI TEKNIKAL MALAYSIA MELAKA,
JALAN HANG TUAH JAYA,
76100 DURIAN TUNGGAL,
MELAKA

Pengalaman Bekerja (Tahun) :

Tandatangan/Cop :

Dr. Mohamad Lutfi Dolhalit
Pensyarah Kanan
Fakulti Teknologi Maklumat & Komunikasi
Universiti Teknikal Malaysia Melaka

Appendix B

Pengesahan Pakar Bidang

Adalah disahkan bahawa kesemua kandungan soalan ujian dan komponen projek (permainan video) yang telah dibina oleh saudara Farhan Akmal bin Yahya (B031810377) dari Fakulti Teknologi Maklumat Komunikasi, Universiti Teknikal Malaysia Melaka, sesuai bagi golongan pelajar dan pekerja yang banyak menggunakan kemahiran menaip.

Sekian, terima kasih.

Ulasan Am:

Konsep yang dibangunkan oleh pelajar adalah bersesuaian untuk tujuan meningkatkan kepantasan dan ketepatan menaip. Namun harus diberi perhatian dari segi UI dan UX dimana ianya perlu dibuat penambahbaikan agar sesuai untuk pengguna/pemain. Pemilihan idea theme dan juga fonts bersesuaian penting untuk memberi pengalaman bermain yang baik kepada pengguna/pemain.

Nama : TS DR HAMZAH ASYRANI SULAIMAN

Jawatan : PENSYARAH KANAN

Tempat Bertugas : UNIVERSITI TEKNIKAL MALAYSIA MELAKA,
JALAN HANG TUAH JAYA,
76100 DURIAN TUNGGAL,
MELAKA

Pengalaman Bekerja (Tahun) : 12

Tandatangan/Cop :

DR. HAMZAH ASYRANI BIN SULAIMA.
Ketua Jabatan
Jabatan Media Interaktif
Fakulti Teknologi Maklumat dan Komunik
Universiti Teknikal Malaysia Melaka (UTe

Appendix C

Sample of Data Link :
https://docs.google.com/spreadsheets/d/1uuUiqntUVM9wQd8nYgFZKeXlSv1uwLJV2itf-FBe0F4/edit?usp=sharing

Game Link :
https://gamejolt.com/games/vocabdefender/629089

Appendix D

Survey Questionnaires

Vocab Defender : A Shooting Typing Game for Typing Skill Improvement

Assalamualaikum and Hi ! Welcome and Thank you for your participation in testing my Final Year Project (FYP) game, Vocab Defender (PC Only). Here is my game's link, https://gamejolt.com/games/vocabdefender/629089

This project is focusing on develop a game that play practice in improving player typing skills and increasing ICT terms usage among players especially students. Vocab Defender is a shooting typing game which its theme is digital world. This game was developed as a typing practice application or tool with a very interesting gameplay and entertaining game world which will increase players' interest in improving their typing skills. Besides that, all the words used in this game are ICT terms that often used which will increase players' ICT terms knowledge indirectly.

Hope players' enjoy this game besides can improve their typing skills and ICT terms usage. Thanks again for joining this project's survey.

* Required

1. Name *

2. Gender *

Mark only one oval.

- ◯ Male
- ◯ Female

3. Age *

Mark only one oval.

- ◯ 7 years old and below
- ◯ 8 - 12 years old
- ◯ 13 - 17 years old
- ◯ 18 - 22 years old
- ◯ 23 - 27 years old
- ◯ 28 - 32 years old
- ◯ 33 years old and above

4. Education Level *

Mark only one oval.

- ◯ Primary School
- ◯ Secondary School
- ◯ University/College
- ◯ Work/Labor

5. If work, state your work field, position and how many years of your work experience (example; Education, ICT Teacher, 10 years). If not work, just fill '-'. *

6. Is your education/work involving or using typing skills? *

Mark only one oval.

- ◯ Yes
- ◯ No

7. If yes, rate how relevant or often you use typing skills in your education/work

Mark only one oval.

	1	2	3	4	5	
Very rarely	◯	◯	◯	◯	◯	Very often

8. Rate your typing skills level *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Very Poor	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very Excellent

9. Which type of video game player is you? *

Mark only one oval.

- ◯ Newbie/novice
- ◯ Casual
- ◯ Midcore/core
- ◯ Hardcore/expert

10. Usual gaming device used? (Can choose more than 1) *

Check all that apply.

- ☐ Computer device (e.g., laptop, desktop)
- ☐ Console device (e.g., Xbox 360, Nintendo Wii)
- ☐ Handheld gaming device (e.g., Game Boy Advance)
- ☐ Mobile device (e.g., smartphone, tablet)
- ☐ Other (e.g., arcade)

11. What the average hours you spent playing game per week? (just approximate) *

Mark only one oval.

- ◯ Less than 1 hr
- ◯ 1–4 hrs
- ◯ 5–9 hrs
- ◯ 10–19 hrs
- ◯ 20–29 hrs
- ◯ 30–39 hrs
- ◯ More than 40 hrs

12. Types of game main genre that you always play. (Can choose more than 1)

Check all that apply.

- ☐ Action (e.g., Destiny, New Super Mario Bros. Wii, Dynasty Warriors 6)
- ☐ Action Adventure (e.g., Minecraft, Grand Theft Auto V, The Last of Us)
- ☐ Driving/racing (e.g., Mario Kart Wii, Forza Horizon 2, Gran Turismo)
- ☐ Fighting (e.g., Super Smash Bros. for Wii U, Skullgirls, Mortal Kombat)
- ☐ Music/dance (e.g., Guitar Hero, Taiko Drum Master, Just Dance 2015)
- ☐ Puzzle/card/board (e.g., Candy Crush Saga, Words With Friends, Tetris)
- ☐ Shooting (FPS/TPS)
- ☐ Role-playing (e.g., World of Warcraft, Dark Souls, Pokemon X)
- ☐ Simulation (e.g., The Sims 3, Space Engineers, Tropico 5)
- ☐ Sports (e.g., NBA 2K15, FIFA 15, SSX)
- ☐ Strategy (e.g., DotA 2, Sid Meier's Civilization V, League of Legends)
- ☐ Trivia/game show (e.g., Trivia Crack, QuizUp, You Don't Know Jack)
- ☐ Other (e.g., Wii Fit, Mario Party, Clicker Heroes)

13. Have you ever played a typing game? *

Mark only one oval.

- ◯ Yes
- ◯ No

14. If yes, state the game's name *

Game User Experience Satisfaction

Player's opinion/feedback after play Vocab Defender game

15. Usability/Playability of the game *

1 = Very Disagree 5 = Very Agree

Mark only one oval per row.

	1	2	3	4	5
I think it is easy to learn how to play the game.					
I find the controls of the game to be straightforward.					
I always know how to achieve my goals/objectives in the game.					
I find the game's interface to be easy to navigate					
I do not need to go through a lengthy tutorial or read a manual to play the game					
I find the game's menus to be user friendly.					
I feel the game trains me well in all of the controls.					
I always know my next goal when I finish an event in the game.					
I feel the game provides me the necessary information to accomplish a goal within the game.					
I think the information provided in the game (e.g., onscreen messages, help) is clear.					
I feel very confident while playing the game.					
I think this game suitable for school and university students					
I think this game suitable for the workers					
I think this game suitable for all ages who want improve their typing skills					
I think the ICT terms used is suitable					

with game theme

16. Play Engrossment/Enjoyment *

1 = Very Disagree 5 = Very Agree

Mark only one oval per row.

	1	2	3	4	5
I think the game is fun.	◯	◯	◯	◯	◯
I enjoy playing the game.	◯	◯	◯	◯	◯
I never feel bored while playing the game.	◯	◯	◯	◯	◯
I think the gameplay and mechanics are interesting	◯	◯	◯	◯	◯
I am likely to recommend this game to others.	◯	◯	◯	◯	◯
If given the chance, I want to play this game again.	◯	◯	◯	◯	◯
I feel detached from the outside world while playing the game	◯	◯	◯	◯	◯
I cannot tell that I am getting tired while playing the game.	◯	◯	◯	◯	◯
I temporarily forget about my everyday worries while playing the game.	◯	◯	◯	◯	◯
I tend to spend more time playing the game than I have planned.	◯	◯	◯	◯	◯
I can block out most other distractions when playing the game.	◯	◯	◯	◯	◯
Whenever I stopped playing the game I cannot wait to start playing it again.	◯	◯	◯	◯	◯

17. Visual and Audio Aesthetics *

1 = Very Disagree 5 = Very Agree

Mark only one oval per row.

	1	2	3	4	5
I enjoy the game's graphics.	◯	◯	◯	◯	◯
I think the game theme and design is interesting	◯	◯	◯	◯	◯
I think the graphics of the game fit the mood or style of the game.	◯	◯	◯	◯	◯
I think the game is visually appealing.	◯	◯	◯	◯	◯
I enjoy the sound effects in the game.	◯	◯	◯	◯	◯
I enjoy the music in the game.	◯	◯	◯	◯	◯
I feel the game's audio (e.g., sound effects, music) enhances my gaming experience	◯	◯	◯	◯	◯
I think the game's audio fits the mood or style of the game	◯	◯	◯	◯	◯

18. Personal Gratification *

1 = Very Disagree 5 = Very Agree

Mark only one oval per row.

	1	2	3	4	5
I am in suspense about whether I will succeed in the game	◯	◯	◯	◯	◯
I feel successful when I overcome the obstacles in the game.	◯	◯	◯	◯	◯
I want to do well as possible during the game.	◯	◯	◯	◯	◯
I am very focused on my own performance while playing the game.	◯	◯	◯	◯	◯
I feel the game constantly motivates me to proceed further to the next stage or level	◯	◯	◯	◯	◯
I find my skills gradually improve through the course of overcoming the challenges in the game	◯	◯	◯	◯	◯
I can measure and know my current typing skill level and its improvement by playing this game	◯	◯	◯	◯	◯
I am sure that I can improve my typing skill if I keep playing this game	◯	◯	◯	◯	◯
This game makes me motivated to keep improving my typing skills	◯	◯	◯	◯	◯
The harder the game's difficulty/challenge, the more I want to do better in improving my typing skills	◯	◯	◯	◯	◯
I am exposed and can identify ICT terms in the game	◯	◯	◯	◯	◯
I am sure that I can improve my ICT terms usage by playing this game	◯	◯	◯	◯	◯
While I keep playing this game, I can detect and memorize some ICT terms used in the game	◯	◯	◯	◯	◯

The gameplay and scoreboard system is interesting and helpful which make me keep playing to improve my typing skills and exposure to the ICT terms

19. Can this game, Vocab Defender, helps you to improve your typing skills and ICT terms exposure? *

20. Why or which part of the game (example : graphics, audio, gameplay, game mechanics, words/terms used, etc) helps you to keep playing and improve your typing skills and ICT terms exposure? Explain your opinion. *

21. What are the advantages of the game in general or in improving players' typing skills? Justify your opinion. (for the expert only)

22. What are the weaknesses of the game in general or in improving players' typing skills? Justify your opinion. (for the expert only)

23. What can be improved for the game to be more effective in improving players' typing skills? (for the expert only)

24. Any overall feedback/suggestions for the game? *

Appendix E - Sample of Source Code

Typer.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Typer : MonoBehaviour
{

    public WordBank wordBank = null;
    public Turret turret;
    public Turret turret1;
    public Turret turret2;
    public Turret turret3;
    public Turret turret4;
    public Turret turret5;
    public Turret turret6;
    public Turret turret7;
    public Turret turret8;
    public Turret turret9;

    public static float masa;
    public static float WPM = 0f;
    public static float Accuracy = 0f;
    private float correctletter = 0f;
    private float totalletter = 0f;
    public static int numword = 0;
    public static int Score = 0;
    private int nummusic = 0;
    public AudioSource fire;
    public AudioSource errorinput;
    public AudioSource music1;
    public AudioSource music2;
    public AudioSource music3;
    public AudioSource music4;
    public AudioSource music5;

    public Text time;
    public Text word;
    public Text wpm;
    public Text score;
    public Text word1;
    public Text wpm1;
    public Text accuracy1;
    public Text score1;
    public Text enemyCount;
    public Text musicoff;

    // A word bank
    public Text wordOutput = null;

    private string remainingWord = string.Empty;
    private string currentWord = string.Empty;

    // Start is called before the first frame update
    private void Start()
    {
        masa = 0;
        WPM = 0f;
```

```
Accuracy = 0f;
correctletter = 0f;
totalletter = 0f;
numword = 0;
Score = 0;
SetCurrentWord();
}

private void SetCurrentWord()
{
    //Get bank word
    currentWord = wordBank.GetWord();
    SetRemainingWord(currentWord);
}

private void SetRemainingWord(string newString)
{
    remainingWord = newString;
    wordOutput.text = remainingWord;
}

// Update is called once per frame
private void Update()
{
    //masa += Time.deltaTime;
    CheckInput();

    wpm.text = string.Format("{0} WPM", WPM.ToString("F2"));
    word.text = string.Format("{0} WORD", numword);
    //accuracy.text = "Accuracy " + Accuracy.ToString("F2") + "%";
    score.text = string.Format("SCORE {0}", Score);
    time.text = string.Format("{00:00}s", masa); //.ToString("F2")
    wpm1.text = string.Format("{0} WPM", WPM.ToString("F2"));
    word1.text = string.Format("{0} WORD", numword);
    accuracy1.text = "Accuracy " + Accuracy.ToString("F2") + "%";
    score1.text = string.Format("SCORE {0}", Score);
    enemyCount.text = string.Format("VIRUS = {0}", WaveSpawner.EnemyAlive);

}

private void CheckInput()
{
    if (WaveSpawner.EnemyAlive > 0)
    {
        if (Input.anyKeyDown)
        {
            string keysPressed = Input.inputString;

            if (keysPressed.Length == 1)
                EnterLetter(keysPressed);
        }
    }
}

private void EnterLetter(string typedLetter)
{
    if (IsCorrectLetter(typedLetter))
    {
        fire.Play();
        RemoveLetter();
        //Turret turret = gameObject.GetComponent<Turret>();
```

```
            correctletter++;
            totalletter++;
            Score++;


            if (IsWordComplete())
            {
                SetCurrentWord();
                numword++;
                turret.Shoot();
                turret1.Shoot();
                turret2.Shoot();
                turret3.Shoot();
                turret4.Shoot();
                turret5.Shoot();
                turret6.Shoot();
                turret7.Shoot();
                turret8.Shoot();
                turret9.Shoot();
            }

        }
        else if (!IsCorrectLetter(typedLetter))
        {
            errorinput.Play();
            correctletter--;
            totalletter++;
        }

        Accuracy = (correctletter / totalletter) * 100;
        WPM = numword / (masa/60);
    }

    private bool IsCorrectLetter(string letter)
    {
        return remainingWord.IndexOf(letter) == 0;
    }

    private void RemoveLetter()
    {
        string newString = remainingWord.Remove(0, 1);
        SetRemainingWord(newString);
    }

    private bool IsWordComplete()
    {
        return remainingWord.Length == 0;
    }

    public void Musicplay()
    {
        if (nummusic > 5)
        {
            nummusic = 0;
        }
        if(nummusic == 0)
        {
            music1.Play();
            music2.Stop();
            music3.Stop();
            music4.Stop();
```

```
            music5.Stop();
            musicoff.text = "ON";
            musicoff.color = Color.green;
        }
        if (nummusic == 1)
        {
            music1.Stop();
            music2.Play();
            music3.Stop();
            music4.Stop();
            music5.Stop();
            musicoff.text = "ON";
            musicoff.color = Color.green;
        }
        if (nummusic == 2)
        {
            music1.Stop();
            music2.Stop();
            music3.Play();
            music4.Stop();
            music5.Stop();
            musicoff.text = "ON";
            musicoff.color = Color.green;
        }
        if (nummusic == 3)
        {
            music1.Stop();
            music2.Stop();
            music3.Stop();
            music4.Play();
            music5.Stop();
            musicoff.text = "ON";
            musicoff.color = Color.green;
        }
        if (nummusic == 4)
        {
            music1.Stop();
            music2.Stop();
            music3.Stop();
            music4.Stop();
            music5.Play();
            musicoff.text = "ON";
            musicoff.color = Color.green;
        }
        if (nummusic == 5)
        {
            music1.Stop();
            music2.Stop();
            music3.Stop();
            music4.Stop();
            music5.Stop();
            musicoff.text = "OFF";
            musicoff.color = Color.red;
        }

        nummusic++;
    }
}
```


Turret.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Turret : MonoBehaviour
{
    //private Bullet bullet;
    private Transform target;

    [Header("Attribute")]

    public float range = 15f;
    public float fireRate = 1f;
    private float fireCountdown = 0f;

    [Header("Unity Setup Fields")]

    public string enemyTag;

    public Transform partToRotate;
    public float turnSpeed = 10f;

    public GameObject bulletPrefab;
    public Transform firePoint;

    [Header("Use Laser")]
    public bool useLaser = false;
    public LineRenderer lineRenderer;


    // Start is called before the first frame update
    void Start()
    {
        InvokeRepeating("UpdateTarget", 0f, 0.5f);
    }

    void UpdateTarget()
    {
        GameObject[] enemies = GameObject.FindGameObjectsWithTag(enemyTag);
        float shortestDistance = Mathf.Infinity;
        GameObject nearestEnemy = null;

        foreach(GameObject enemy in enemies)
        {
            float distanceToEnemy = Vector3.Distance(transform.position,
enemy.transform.position);
            if(distanceToEnemy < shortestDistance)
            {
                shortestDistance = distanceToEnemy;
                nearestEnemy = enemy;
            }
        }

        if (nearestEnemy != null && shortestDistance <= range)
        {
            target = nearestEnemy.transform;
        }
        else
        {
            target = null;
        }
```

```
    }

    // Update is called once per frame
    void Update()
    {
        if (target == null)
        {
            if(useLaser)
            {
                if (lineRenderer.enabled)
                    lineRenderer.enabled = false;
            }

            return;
        }

        //Target lock on
        Vector3 dir = target.position - transform.position;
        Quaternion lookRotation = Quaternion.LookRotation(dir);
        Vector3 rotation = Quaternion.Lerp(partToRotate.rotation, lookRotation,
Time.deltaTime * turnSpeed).eulerAngles;
        partToRotate.rotation = Quaternion.Euler(rotation.x, rotation.y,
rotation.z);

        if (useLaser)
        {
            Laser();
        }
        else
        {
            if (fireCountdown <= 0f)
            {
                fireCountdown = 1f / fireRate;
            }
            fireCountdown -= Time.deltaTime;
        }

    }
    public void Shoot()
    {

        Debug.Log("SHOOT!");
        GameObject bulletGO = (GameObject)Instantiate(bulletPrefab,
firePoint.position, firePoint.rotation);
        Bullet bullet = bulletGO.GetComponent<Bullet>();

        if (bullet != null)
        {
            bullet.Seek(target);
        }


    }

    public void Laser()
    {
        if (!lineRenderer.enabled)
            lineRenderer.enabled = true;

        lineRenderer.SetPosition(0, firePoint.position);
        lineRenderer.SetPosition(1, target.position);
```

```
    }

    void OnDrawGizmosSelected()
    {
        Gizmos.color = Color.red;
        Gizmos.DrawWireSphere(transform.position, range);
    }
}
```

WaveSpanwer

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.Runtime.InteropServices;

public class WaveSpawner : MonoBehaviour
{

    public static int EnemyAlive = 0;
    public static int round = 0;
    //public GameObject capsulehealth;
    public int bil = 0;
    public int allowsideturret = 0;
    public GameObject otherturret;
    public GameObject otherturret1;
    public GameObject capslockon;
    public GameObject capslockoff;
    private bool capsison;

    public Wave wave;

    public Transform enemyPrefab;
    public Transform enemyPrefab1;
    public Transform spawnPoint;
    public Transform spawnPoint1;

    public Vector3 center;
    public Vector3 size;
    public Vector3 center1;
    public Vector3 size1;
    public Vector3 center2;
    public Vector3 size2;
    public Vector3 center3;
    public Vector3 size3;
    public Vector3 center4;
    public Vector3 size4;
    public Vector3 center5;
    public Vector3 size5;
```

```
public Vector3 center6;
public Vector3 size6;
public Vector3 center7;
public Vector3 size7;
public Vector3 center8;
public Vector3 size8;


public float timeBetweenWaves = 10f;
private float countdown = 10f;

public Text waveCountDownText;
public Text Round;
public Text Round1;

private int waveIndex = 0;

public void Start()
{
   //isCapsLockOn = (((ushort)GetKeyState(0x14)) & 0xffff) != 0;//init stat

   EnemyAlive = 0;
   round = 0;
   Round.text = "ROUND " + round;
   Round1.text = "ROUND " + round;
   //capslockoff.SetActive(false);
   //capslockon.SetActive(true);
   capsison = true;
}

/*void OnGUI()
{
   Event e = Event.current;
   if (e.capsLock)
   {
      GUI.Label(new Rect(10, 10, 100, 20), "CapsLock on.");
      StartCoroutine(DisappearCapsLock());

      capsison = true;
   }
   else
   {
      GUI.Label(new Rect(10, 10, 100, 20), "CapsLock off.");
      capslockoff.SetActive(true);
      capslockon.SetActive(false);
      capslockon.transform.localScale = new Vector3(1, 1, 1);

      capsison = false;
   }
}*/
```

```
    void Update()
    {
        //if (Input.GetKeyDown(KeyCode.CapsLock))
        //{
        //isCapsLockOn = !isCapsLockOn;
         /* if (GetCapsLock() == false) //capslockisoff
            {
               capslockoff.SetActive(true);
               capslockon.SetActive(false);
            capslockon.transform.localScale = new Vector3(1, 1, 1);

            capsison = false;
            }
            if (GetCapsLock() == true) //capslockison
            {
               StartCoroutine(DisappearCapsLock());

               capsison = true;
            }*/

        //}

        if (capsison == true)
        {


            if (EnemyAlive > 0)
            {
               Typer.masa += Time.deltaTime;
               return;
            }

            if (countdown <= 0f)
            {

               waveCountDownText.color = Color.white;
               StartCoroutine(SpawnWave());
               StartCoroutine(SpawnWave1());
               StartCoroutine(Spawnwave());
               countdown = timeBetweenWaves;
               round += 1;
               Round.text = "ROUND " + round;
               Round1.text = "ROUND " + round;
               bil += 1;
               if (bil == 3)
               {
                  if (PlayerHealth.currenttowerhealth >= 1 &&
PlayerHealth.currenttowerhealth <= 120)
                     PlayerHealth.currenttowerhealth += 400;
```

```
            if (PlayerHealth.currenttowerhealth > 120 &&
PlayerHealth.currenttowerhealth <= 240)
                PlayerHealth.currenttowerhealth += 300;
            if (PlayerHealth.currenttowerhealth > 240 &&
PlayerHealth.currenttowerhealth <= 360)
                PlayerHealth.currenttowerhealth += 200;
            if (PlayerHealth.currenttowerhealth > 360 &&
PlayerHealth.currenttowerhealth <= 480)
                PlayerHealth.currenttowerhealth += 100;
            if (PlayerHealth.currenttowerhealth > 480 &&
PlayerHealth.currenttowerhealth <= 500)
                PlayerHealth.currenttowerhealth += 50;
            if (PlayerHealth.currenttowerhealth > 500 &&
PlayerHealth.currenttowerhealth <= 600)
                PlayerHealth.currenttowerhealth += 0;
            bil = 0;
        }
        allowsideturret += 1;
        if (allowsideturret >= 5)
        {
            otherturret.SetActive(true);
        }
        if (allowsideturret >= 15)
        {
            otherturret1.SetActive(true);
        }

        return;
    }

    countdown -= Time.deltaTime;

    countdown = Mathf.Clamp(countdown, 0f, Mathf.Infinity);

    waveCountDownText.text = string.Format("{00:00}", countdown);
    waveCountDownText.color = Color.red;


    if (EnemyAlive <= 0)
    {
        EnemyAlive = 0;
    }
  }
}

IEnumerator DisappearCapsLock()
{
  capslockoff.SetActive(false);
  capslockon.SetActive(true);
```

```
    if (capsison == true)
    {
        yield return new WaitForSeconds(1.5f);
        capslockon.transform.localScale = new Vector3(0, 0,0);
    }
}

IEnumerator SpawnWave()
{
    //waveIndex++;
    waveIndex += 1;

    //Wave wave = waves[waveIndex];

    //Debug.Log("Wave Incoming!");
    for (int i = 0; i < waveIndex; i++)
    {
        Spawnenemy();
        yield return new WaitForSeconds(1f);

    }

    /*if (waveIndex == waves.Length)
    {

    }*/
}

void Spawnenemy()
{
    Instantiate(enemyPrefab, spawnPoint.position, enemyPrefab.rotation);
    Instantiate(enemyPrefab1, spawnPoint1.position, enemyPrefab1.rotation);
    EnemyAlive++;
    EnemyAlive++;
}

IEnumerator SpawnWave1()
{
    for (int i = 0; i < waveIndex; i++)
    {
        Spawnenemy1(wave.enemy);
        yield return new WaitForSeconds(0.5f);

    }
}

void Spawnenemy1(GameObject enemy)
{
    Instantiate(enemy, spawnPoint.position, enemyPrefab.rotation);
```

```
    EnemyAlive++;
}

IEnumerator Spawnwave()
{
    //waveIndex++;
    //waveIndex += 5;

    //Debug.Log("Wave Incoming!");
    for (int i = 0; i < waveIndex; i++)
    {
        SpawnObject(wave.enemy2);
        SpawnObject1(wave.enemy2);
        SpawnObject2(wave.enemy2);
        SpawnObject3(wave.enemy2);
        SpawnObject4(wave.enemy3);
        SpawnObject5(wave.enemy3);
        SpawnObject6(wave.enemy3);
        SpawnObject7(wave.enemy3);

        yield return new WaitForSeconds(0.5f);

    }

    //SpawnObject8();
}


void OnDrawGizmosSelected()
{
    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center, size);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center1, size1);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center2, size2);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center3, size3);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center4, size4);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center5, size5);

    Gizmos.color = new Color(1, 0, 0, 0.5f);
    Gizmos.DrawCube(center6, size6);
```

```
        Gizmos.color = new Color(1, 0, 0, 0.5f);
        Gizmos.DrawCube(center7, size7);

        Gizmos.color = new Color(1, 0, 0, 0.5f);
        Gizmos.DrawCube(center8, size8);
    }

    public void SpawnObject(GameObject enemy2)
    {
        Vector3 pos = center + new Vector3(Random.Range(-size.x / 2, size.x / 2), 1,
Random.Range(-size.z / 2, size.z / 2));

        Instantiate(enemy2, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject1(GameObject enemy2)
    {
        Vector3 pos = center1 + new Vector3(Random.Range(-size1.x / 2, size1.x / 2), 1,
Random.Range(-size1.z / 2, size1.z / 2));

        Instantiate(enemy2, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject2(GameObject enemy2)
    {
        Vector3 pos = center2 + new Vector3(Random.Range(-size2.x / 2, size2.x / 2), 1,
Random.Range(-size2.z / 2, size2.z / 2));

        Instantiate(enemy2, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject3(GameObject enemy2)
    {
        Vector3 pos = center3 + new Vector3(Random.Range(-size3.x / 2, size3.x / 2), 1,
Random.Range(-size3.z / 2, size3.z / 2));

        Instantiate(enemy2, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject4(GameObject enemy3)
    {
```

```
        Vector3 pos = center4 + new Vector3(Random.Range(-size4.x / 2, size4.x / 2), 1,
Random.Range(-size4.z / 2, size4.z / 2));

        Instantiate(enemy3, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject5(GameObject enemy3)
    {
        Vector3 pos = center5 + new Vector3(Random.Range(-size5.x / 2, size5.x / 2), 1,
Random.Range(-size5.z / 2, size5.z / 2));

        Instantiate(enemy3, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject6(GameObject enemy3)
    {
        Vector3 pos = center6 + new Vector3(Random.Range(-size6.x / 2, size6.x / 2), 1,
Random.Range(-size6.z / 2, size6.z / 2));

        Instantiate(enemy3, pos, Quaternion.identity);

        EnemyAlive++;
    }

    public void SpawnObject7(GameObject enemy3)
    {
        Vector3 pos = center7 + new Vector3(Random.Range(-size7.x / 2, size7.x / 2), 1,
Random.Range(-size7.z / 2, size7.z / 2));

        Instantiate(enemy3, pos, Quaternion.identity);

        EnemyAlive++;
    }
}
```

PlayerHealth.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerHealth : MonoBehaviour
{
    public PlayFabManager playfabManager;
    public int score;
    private int wpm;
    private int accuracy;
```

```
    public static float currenttowerhealth;
    public GameObject gameoverlayout;
    public GameObject ingamelayout;

    public GameObject camera1;
    public GameObject camera2;
    private int nocamera = 0;

    public void Start()
    {
        ingamelayout.SetActive(true);
        gameoverlayout.SetActive(false);
        currenttowerhealth = 600;
        nocamera = 1;
    }
    public void Update()
    {
        if (currenttowerhealth <= 0)
        {
            GameOver();
        }

        if(Input.GetKey(KeyCode.Escape))
        {
            GameOver();
        }

    }

    public void GameOver()
    {
        ingamelayout.SetActive(false);
        gameoverlayout.SetActive(true);
        //Time.timeScale = 0;
        score = Typer.Score;
        wpm = (int)Typer.WPM;
        accuracy = (int)Typer.Accuracy;
        playfabManager.SendLeaderboard(score, Typer.numword, wpm, accuracy,
WaveSpawner.round);
    }

    public void ChangeCamera()
    {
        if(nocamera > 2)
        {
            nocamera = 1;
        }
        if (nocamera == 1)
        {
            camera1.SetActive(false);
            camera2.SetActive(true);
        }
        if(nocamera == 2)
        {
            camera1.SetActive(true);
            camera2.SetActive(false);
        }


        nocamera++;
    }
}
```