

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

**A Top-down action game for learning English with
Unity**

Vũ Hải Long

long.vh176041@sis.hust.edu.vn

Major: Information Technology

Supervisor: Phd. Nguyễn Tuân Dũng

Signature

Department: Computer Engineering

School: School of Information and Communications Technology

HANOI, 08/2022

ACKNOWLEDGMENT

First of all, I want to thank PhD. Nguyen Tuan Dung, who was my primary tutor. Your assistance is crucial for me to finish this project. Thanks to your direction, I have also learnt a lot, and the wealth of knowledge I have gained will be very helpful to me as I pursue my dream of becoming a great programmer. Additionally, I would want to express my gratitude to the Hanoi University of Science and Technology in general and to all instructors that have helped me learn and grow over the past five years in particular. I have improved both professionally and morally as a result of the knowledge and lessons I have learned, which are of incalculable value. My last thanks go to my family and friends. They have always supported me and are also a great source of motivation for me to perfect this product.

ABSTRACT

Technology is developing at a breakneck speed and the world is becoming more and more open. Needless to say too much about the importance of integration and learning, absorbing the quintessence and progress from other cultures. To do that, we probably all know how important a foreign language is. Learning a foreign language means learning more about a culture and acquiring new knowledge. There are a lot of languages we can choose from based on our preferences and interests, but we can all agree that English is one of the best choices. Currently, English is the official language of more than 53 countries and territories, the second language of nearly 100 countries, and the common language of more than 400 million people around the world. In this era, almost all professions of commerce, health, media, tourism, etc. use English. At most meetings, forums, exchanges of work, knowledge and experiences around the world, this international language is used as a bridge. That is why in the past 10 years in Vietnam, English centers have appeared and developed very strongly. Technology is also used to create programs and applications for learning English that are very popular and has had obvious effects. As an avid language learner and also an Information technology student who likes to play video games, I've always had ideas about developing a game that can help me learn English effectively. Before deciding whether to realize the above idea, I had certain research about the game industry and realized its great attraction at the present time. Therefore, I have decided that my graduation project is a game that can help users learn English with the expectation that my product will help users be both entertaining and educational.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION.....	1
1.1 Overview of the Game Industry	1
1.2 Overview of the thesis.....	3
1.3 Game story	4
CHAPTER 2. REQUIREMENT SURVEY AND SYSTEM ANALYSIS ...	7
2.1 Survey	7
2.2 System analysis	7
2.2.1 Class diagram.....	7
2.2.2 Sequence diagram	9
2.2.3 Activity diagram.....	11
CHAPTER 3. GAME TECHNOLOGY AND DESIGN.....	15
3.1 Game technology.....	15
3.2 Game design	17
3.2.1 Core gameplay	17
3.2.2 Game loop.....	18
3.2.3 Game Unit.....	18
3.2.4 Enemy AI.....	29
3.2.5 Vocabulary system	29
3.2.6 UI Flow.....	30
3.2.7 A few main UI	30
CHAPTER 4. EXPERIMENT AND EVALUATION.....	34
4.1 Test case	34
4.1.1 Control character movement	34
4.1.2 Control character shoot	34

4.1.3 Cast character skill 1	34
4.1.4 Cast character skill 2	35
4.1.5 Cast character skill 3	35
4.1.6 Cast character skill 4	35
4.1.7 Check character and enemy collision	36
4.1.8 Check character die	36
4.1.9 Check enemy and bullet collision	36
4.1.10 Check upgrade system	36
4.1.11 Check high score system	37
4.1.12 Check setting system	37
4.1.13 Check vocabulary system	37
4.2 Evaluation.....	37
CHAPTER 5. CONCLUSION	39
5.1 Solution and Contribution	39
5.1.1 Game graphic and sound.....	39
5.1.2 Scalability	39
5.1.3 Performance	40
5.2 Conclusion and Future Work	41

LIST OF FIGURES

Figure 2.1	Class diagram	8
Figure 2.2	Sequence diagram of Setting	9
Figure 2.3	Sequence diagram of Highscore	10
Figure 2.4	Sequence diagram of Start gameplay	10
Figure 2.5	Sequence diagram of Upgrade	10
Figure 2.6	Sequence diagram of handle gameloop	11
Figure 2.7	Activity diagram of Setting feature	12
Figure 2.8	Activity diagram of Highscore feature	12
Figure 2.9	Activity diagram of Start gameplay feature	13
Figure 2.10	Activity diagram of Upgrade feature	13
Figure 2.11	Activity diagram of handle gameloop feature	14
Figure 3.1	Player's skill 1	18
Figure 3.2	Player	19
Figure 3.3	Player's skill 1	20
Figure 3.4	Player's skill 2	21
Figure 3.5	Player's skill 3	22
Figure 3.6	Player's skill 4	23
Figure 3.7	Monster	24
Figure 3.8	House	25
Figure 3.9	Well	26
Figure 3.10	Fence	27
Figure 3.11	Buff	28
Figure 3.12	Enemy state machine	29
Figure 3.13	UI flow	30
Figure 3.14	Start screen	30
Figure 3.15	Highscore screen	31
Figure 3.16	Inventory screen	32
Figure 3.17	Choose map screen	32
Figure 3.18	Gameplay screen	33
Figure 4.1	Test case for control character movement	34
Figure 4.2	Test case for control character shoot	34
Figure 4.3	Test case for cast character skill 1	35
Figure 4.4	Test case for cast character skill 2	35
Figure 4.5	Test case for cast character skill 3	35

Figure 4.6	Test case for cast character skill 4	36
Figure 4.7	Test case for check character and enemy collision	36
Figure 4.8	Test case for check character die	36
Figure 4.9	Test case for check enemy and bullet collision	37
Figure 4.10	Test case for check upgrade system	37
Figure 4.11	Test case for check high score system	37
Figure 4.12	Test case for check setting system	38
Figure 4.13	Test case for check vocabulary system	38

CHAPTER 1. INTRODUCTION

1.1 Overview of the Game Industry

From the heyday of arcade machines to the rise of home consoles, the gaming industry has grown exponentially over the past few decades. We can divide the development of the game industry into several different stages.

1970–1983: Pre-crisis era Atari is a name that should be mentioned in the history of the game industry. Early prototypes of video games were developed in a lab in the 1960s, but it was Atari's release of Pong in 1972 that helped kickstart the industry. The arcade version of table tennis is an attractive game, attracting customers eager to play and companies began to produce their clones. Likewise, Atari himself sold a version of the game for Pong's console in 1975 and finally the Atari 2600-branded console in 1977, which would be the first console to sell more than a single console million units. In a short time, the arcade market began to stabilize. After the game market declined due to too many copies of Pong, the release of Space Invaders in 1978 revived the market. Arcade machines began to be installed everywhere and new series of games like Pac-Man and Donkey Kong have promoted the further development of this type of entertainment. By 1982, arcade games were generating more money than both the pop industry and the box office.

1985–2000: The race for technological progress. Unfortunately, the pace of development of the gaming industry is too fast to sustain. Eager to profit from the burgeoning (household) console game market, Atari licensed the ultra-budget Pac-Man docking stations and an adaptation of the E.T. the Extra-Terrestrial. They hit the market in a hurry. As a result, the poor quality of the game cost the company millions of dollars in profits and brand damage. As other companies also sought to capitalize on the market, many poor-quality console and game products caused an industry-wide downturn. At the same time, the personal computer became the newest gaming machine, especially with the release of the Commodore 64 in 1982. It was a defining sign of the historical era of the gaming industry: a run technology race. In the years that followed, Nintendo released the Nintendo Entertainment System (NES) console in 1985 (released in Japan as the Famicom), prioritizing high-quality games and consistent marketing to recapture the market. The market is feeling cautious. Thanks to games like Duck Hunt, Excitebike, and the introduction of Mario in Super Mario Bros, the NES' huge success revived the console market. Nintendo wanted to continue its dominance in the field with the release of the Game Boy handheld and the Super Nintendo entertainment system. In 1988, the

Sega arcade company entered the fray with the Sega Mega Drive console (released as the Sega Mega Drive console called Genesis in North America) and later the Game Gear handheld, placing a marketing emphasis on processing power. Electronic manufacturer Sony released the PlayStation in 1994, using CD-ROM discs instead of electronic tapes to increase storage capacity for individual games. It became the first console in history to sell over 100 million units, and the focus on the software format continued with the next generation of consoles: PlayStation 2 (DVD) and PlayStation 3 (Blu-ray). Even Microsoft recognized the importance of PC gaming and developed the DirectX API to aid in game programming. That "X" brand later helped the company break into the console market with the Xbox product.

2001–now: The online boom. However, it is the rise of the Internet and mobile phones that has taken the gaming industry from tens of billions to hundreds of billions of dollars in revenue. The main reason is thanks to the appearance of subscription services (subscriptions) and freemium services. The company with a freemium model, a combination of “free” and “premium”, provides basic services initially for free to users but to use more premium services or extra features, customers need to pay. In 2001, Microsoft launched the Xbox Live online gaming platform with a monthly subscription, giving players access to voice chat and multiplayer connectivity services. It quickly became a must-have feature for consumers. Meanwhile, on PC, Blizzard tapped into the massively multiplayer online (MMO) subscription market with the 2004 release of World of Warcraft, which peaked at over 14 million monthly paying subscribers. At the same time, the companies saw a bright future for mobile games, but they encountered many difficulties in the exploitation process. Nintendo continued to dominate the handheld market with updated Game Boy releases, and Nokia and BlackBerry tried integrating game apps into their phones. But it was Apple’s iPhone that underpinned the game’s transition to mobile platforms. The company’s release of its App Store for smartphones (followed by Google’s store for Android devices) paved the way for app developers to create free, paid games, and pay-per-feature to serve the mass market. Now, everyone’s eyes are on that burgeoning 85 billion dollars mobile gaming market, and game companies are beginning to solidify their positions in this segment. Console makers like Microsoft and Sony are rolling out cloud subscription services while they continue to develop new consoles. Meanwhile, Amazon and Google are rolling out their services that work across multiple devices, including mobile devices. After seeing the success that games like Pokemon Go have on smartphones, the title hit over 1 billion dollars in annual sales and broke Grand Theft Auto V’s 1

billion dollar sales record in just over a year and three days, companies are trying to capture as much share of the mobile gaming market as possible. With the proliferation of smartphones, social gaming and streaming services, they're on the right track. There are over 2.7 billion people playing games worldwide in 2020, and how they choose to spend their money will continue to shape the history of the gaming industry.

From the above information, we can see that the game industry is a potential market which will grow a lot in the future, and also has a great attraction to the young generation.

1.2 Overview of the thesis

As mentioned above, this project has the main purpose of helping players learn English. This game will focus on helping players learn vocabulary. The reason is that vocabulary is the "root source" of all problems. To communicate coherently and capture the necessary information with the other party, vocabulary is what you need to be sure of. Only when possessing a rich and rich vocabulary can your speech become natural and fluent. In the process of communication, readers or listeners almost only pay attention to the content that the speaker wants to express rather than grammar and sentences. Because, even if you firmly grasp the grammatical structures, the vocabulary is limited, it is still impossible to properly convey the content that you want to convey to the listener. Rich and abundant vocabulary will greatly assist you in developing and perfecting other skills more effectively. With a wide range of vocabulary, you can easily understand the conveyed content even if your grammar is not too strong. And when you understand quickly and get it right, you will be able to respond quickly. The process of listening, understanding and speaking the word will also become more and more fluent. To help players learn vocabulary systematically from easy to difficult and acquire knowledge more quality and effective, this project did classify the difficulty of vocabulary and apply it to the gameplay. The difficulty scale this essay decided to apply as a standard is called CEFR. CEFR stands for 'Common European Framework of Reference for languages, which describes the language abilities of students at different levels of learning. CEFR can be used to compare standards in language learning and to create instructional programs. It ranks language skills at six levels. A1 and A2 indicate elementary and elementary levels of ability, B1 and B2 indicate lower and upper intermediate, C1 indicates advanced level, C2 indicates complete proficiency in the language.

Based on the above reference scale, the game is divided into 5 levels with diffi-

culty levels ranging from A1 to C1. To complete the game, an important factor is that the player must continuously learn the vocabulary at each level to a sufficient level of proficiency, which will stimulate the player to absorb new knowledge in the process of experiencing. Due to a large number of English words, to avoid the loss of knowledge, the vocabulary data set has been selected the most important words (words with a lot of frequency) at each level of vocabulary to optimize learning efficiency. This dataset is based on two main books, Oxford 3000 and Oxford 5000. As for the gameplay, to increase the attractiveness and continuity of the players, I decided to choose action-oriented game development. The viewing angle of the game is a Top-down view so that players do not have vestibular problems such as dizziness, and headaches when playing games for a long time. The goal of the project is that players can enjoy the feeling of playing the game and learn a large number of vocabularies.

1.3 Game story

The plot is the "backbone", creating the "soul" for a game, helping to push the emotions when experiencing and contributing to the rationalization of gameplay. The following is the plot for this game product. This is the story of continent A, an isolated region that has never been explored. This world is a place where magic is supreme and is ruled by two empires B and C. To explain the origin of these two empires, one must dive into the deepest depths of the river of history here. At that time, the methods of survival were very instinctive and minimalist while nature was extremely harsh. Thus, this was a very dark period in human history that is later often referred to as the "Age of Illiteracy". Humans continued to live in disarray until one day, on dry soil in Continent A's eastern half, there was a boy who was suffering the most painful moments of his life since he was born. His family had just been slaughtered by a swarm of insects while trying to find food to survive the meteorite season. He is the last surviving member of the family but is also surrounded by a huge swarm of cockroaches. Realizing that he no longer had a chance to survive, the memory of his family becoming food for the insects that suddenly appeared, pushed his mind to the brink of collapse. He raised his face to the sky and shouted to release all his resentment towards life, but due to the pain and exhaustion, he couldn't scream like a normal person. His mouth was constantly deformed, his breathing was also constantly interrupted, which inadvertently made his screams become a strange sound that had never appeared before. He shouted until his throat was torn, his strength and strength exhausted and he gave up hope of living, suddenly his body felt indescribable as if he was connected to this continent. This strange feeling pulled him out of the pit of despair, made him look around and

this scene was forever imprinted in his mind until the last days of his life. The story of the boy in a desperate scream for extermination who unleashed the power to destroy the cockroaches has quickly spread throughout the A continent. Some believe the boy has special powers, others say that the boy's scream had touched a higher-level existence, some just considered it a coincidence, etc. But whatever the conclusion, countless people tried to research its usage. That power is based on rumors, they believe that as long as they control this power, they will become the hegemony of the continent. It has been proven that this power is indeed attainable when in the following years it has begun to appear that some people can create phenomena beyond human understanding when trying to understand them. making strange sounds with the mouth. This reinforced the belief that there was a power beyond human ability and from this point on, the entire A continent was immersed in trying to control this power. Over the years, the research and application of this mysterious power have come a long way. The sound that the boy made this year is known as an ancient language by humans, this is also the reason why the quality of life in Continent A has grown to a height beyond all imagination. Ancient languages occupy absolute importance in life here, appearing in all fields such as military, mining, education, economy, etc. Through countless years of research and development, the world has only discovered 2 types of ancient languages, called English and Vietnamese. Each type has its characteristics but all bring great benefits in all aspects of life. A special feature is that each person can only use one type of ancient language - that is the first type of ancient language they use in their life. This has inadvertently created two opposing factions in Continent A when every user of the ancient language believes that the language they use is the original strength of this continent. This ideological conflict over time splits the people living here into two main factions, empire B and C. At first, political conflicts between B and C took place extremely tensely when the two empires were very tense. This nation competes on all fronts to prove its rune superior. However, over time, this conflict gradually cooled down when neither side can take a clear advantage and the power of the two empires is always in balance. People also gradually believe that the power of the two ancient languages is equivalent and the contempt for those who use the other languages gradually disappears, the people of the two empires live in peace, and the people of the mainland live together in peace. Land A gradually became prosperous and developed. It was thought this peace would last forever, but an event that took place some 200 years ago developed the seeds that threatened to destroy them all. A genius researcher discovered a way to combine two ancient languages, English and Vietnamese, when he accidentally discovered

that these two ancient languages had words with the same meaning. He also realized when combining these two words, he will gain an unsurpassed supreme power. Excited by this discovery but also aware of the huge impact it would have on him, he spent the rest of his life searching for all the words with the same meaning in English and Vietnamese alone, then record all that knowledge in a book. This book was named "English-Vietnamese Dictionary" by him, then he hid this book in a very secret place and intended never to publish this research. However, because he regretted his efforts, he told this to his son and warned that this was a family secret not to be revealed to anyone before he die. This secret was kept hidden only by a single member of the family until D's lifetime. In a world where ancient languages were so essential, one's social status was determined by one's ability to use ancient languages. D is a poor student who always struggles with the most basic words despite trying hard to learn, leading to D being bullied and despised by his friends. This took place and built up in D an ideal that wanted to change the world where no one had to learn a language by creating a new language that people could still use without having to learn. He decided to find the dictionary revealed by his father in the last days of his life. After an arduous journey, D successfully finds the dictionary and tries to use it to change the rules of the world. However, due to a lack of basic knowledge of grammar, he accidentally combined the incantation into extremely strange and hybrid sounds between the two ancient languages. This spell overturned all the principles of English and Vietnamese, causing these two ancient languages to become chaotic and completely disable all their power. This caused an unprecedented crisis on the A continent when it threatened to bring it back to the Age of Illiteracy. Mankind's only hope is to collect all the words that have become chaotic in the English-Vietnamese dictionary. Regretting his behavior, D decides to go back and collect the lost word pairs in the dictionary. Please use your English knowledge to help D redeem his mistake!

CHAPTER 2. REQUIREMENT SURVEY AND SYSTEM ANALYSIS

2.1 Survey

As stated in the abstract, there is currently a very high demand for English instruction in Vietnam. Although the English training market has recently been heavily mined, there is still a ton of untapped potential, particularly in the area of English learning software development. Applications for learning English like Duolingo, LingoDeer, Cake, and others have proven to be quite popular and useful for users. There are a few applications that have integrated games to increase the fun and excitement in the learning process. However, it can be said that the target audience of most applications containing games is children who have just started learning foreign languages when the gameplay is very simple and easy to play. This is very suitable for young children but will not be very effective for those of an older age when the monotonous gameplay will quickly become boring.

Based on the fact that the market is lacking in English learning games for teenagers, this will be the main user group that this project is aimed at. Gameplay will be designed quickly and with dramatic elements, causing high difficulty and requiring that in addition to the player having English knowledge, they must also be able to control the character skillfully to complete the game.

2.2 System analysis

2.2.1 Class diagram

Class diagram Figure 2.1 shows the relationship between objects in the process of building the system of this project. Some important classes that have a lot of influence are GameManager, PlayerManager, DataManager, EnemyManager, etc. As soon as the game starts, the DataManager class will be called to reload the data from the player's last session. After the player selects the screen in CanvasChooseMap, the system will load the map data in LoadingManager. Once the loading is complete, the GameManager will receive the signal and trigger all the necessary classes to start performing their tasks to start the gameplay. Triggered classes include SpawnerManager, BuffManager, PlayerManager, and GameManager also calls CanvasGameplay to appear. When the GameManager receives the signal to end the game when the player wins or loses, the GameManager will call the DataManager to save the necessary information and return the UI to CanvasChooseMap.

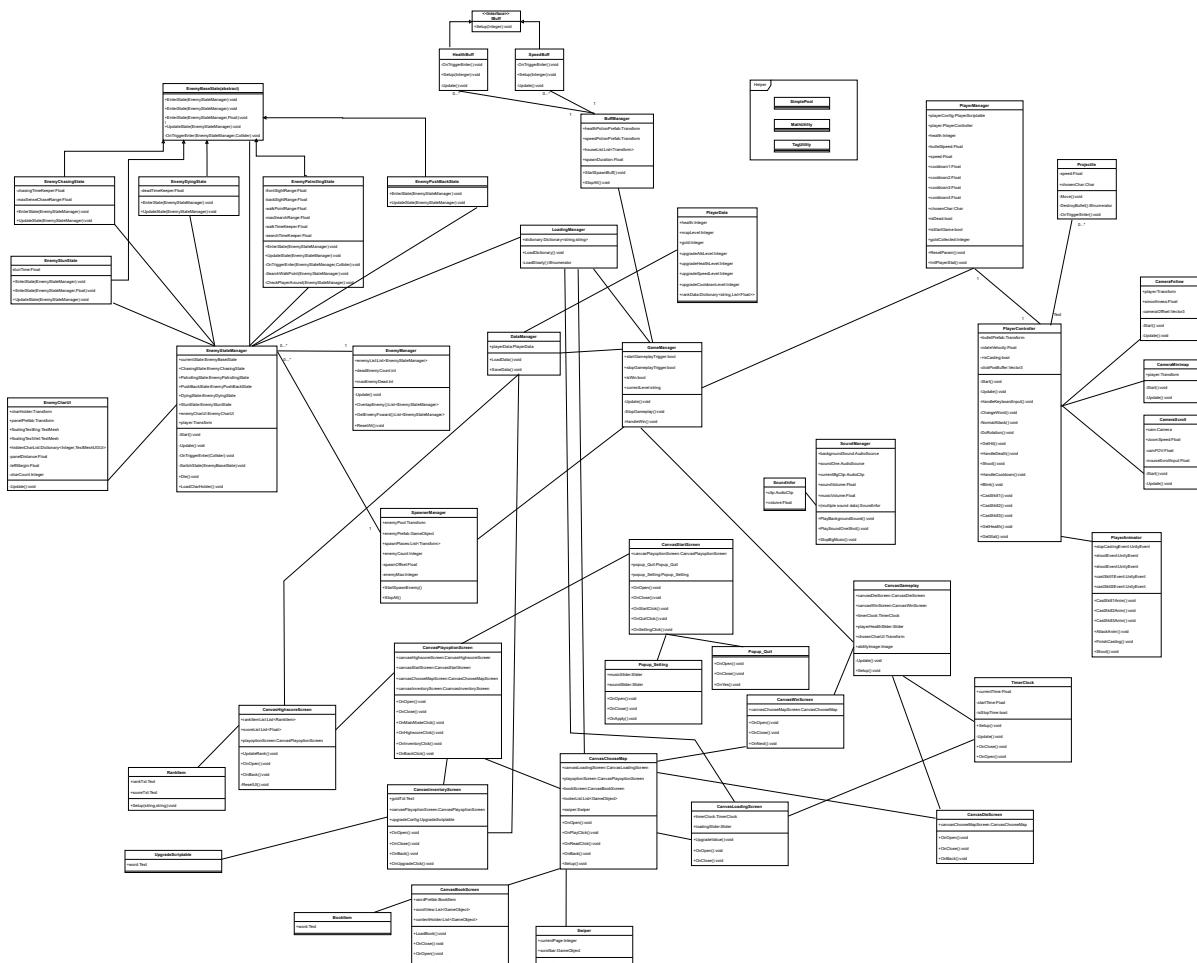


Figure 2.1: Class diagram

2.2.2 Sequence diagram

Sequence diagram is a diagram used to identify the sequence of events of a certain group of objects. It describes in detail the messages sent and received between objects and also focuses on the timing of sending and receiving those messages. The following are the main sequence diagrams of the system.

a, Setting

Figure 2.2 is the sequence diagram of the Setting feature. The flow starts when the player clicks the Setting button in the Canvas Start Screen. After the adjustment is complete, the game's volume will be updated again in Sound Manager.

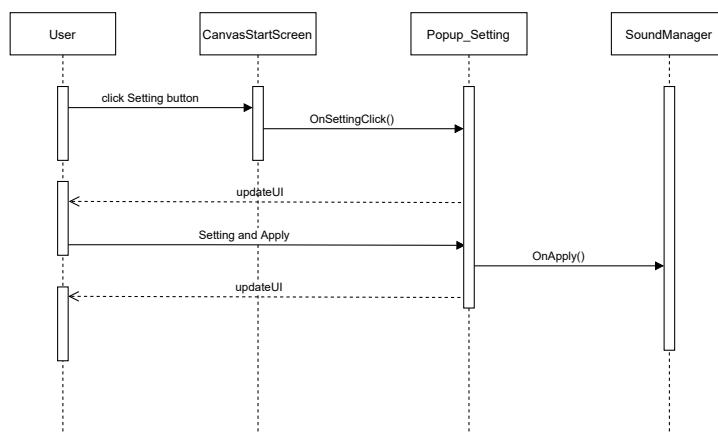


Figure 2.2: Sequence diagram of Setting

b, Highscore

Figure 2.3 is the sequence diagram of Highscore feature. The stream will start when the player clicks on the High score button in the Canvas Play option Screen. The system will access the Data Manager to get the data of the gameplay completion time and display it back to the Canvas High Score Screen.

c, Start

Figure 2.4 is the sequence diagram of Start gameplay feature. After the player presses Play on the Canvas Choose Map and the Loading Manager finishes loading the data, the GameManager will trigger a few other classes to start the main gameplay.

d, Upgrade

Figure 2.5 is the sequence diagram of Upgrade feature. The flow of this diagram revolves around the DataManager so that it can check if the player has enough resources to upgrade.

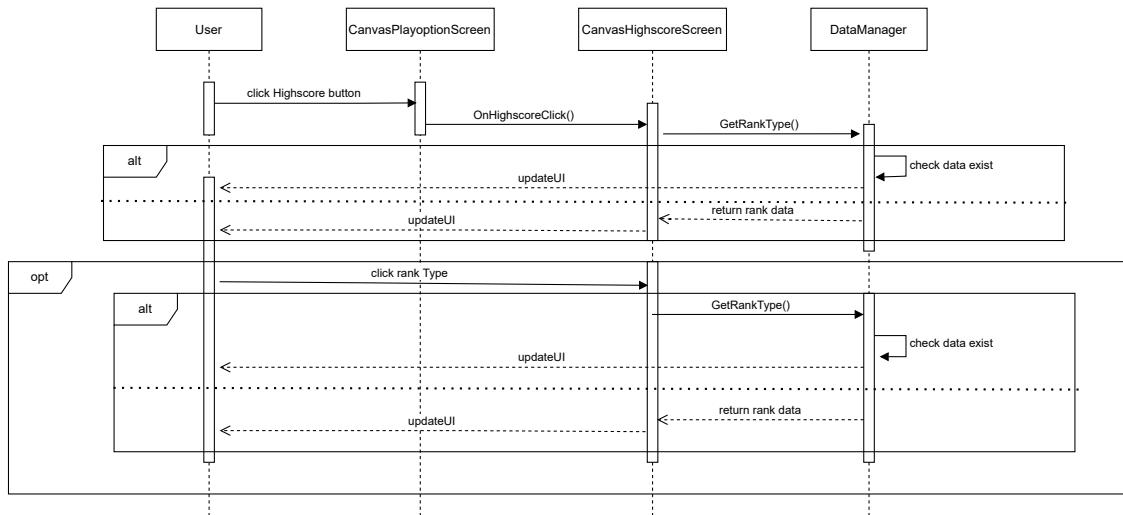


Figure 2.3: Sequence diagram of Highscore

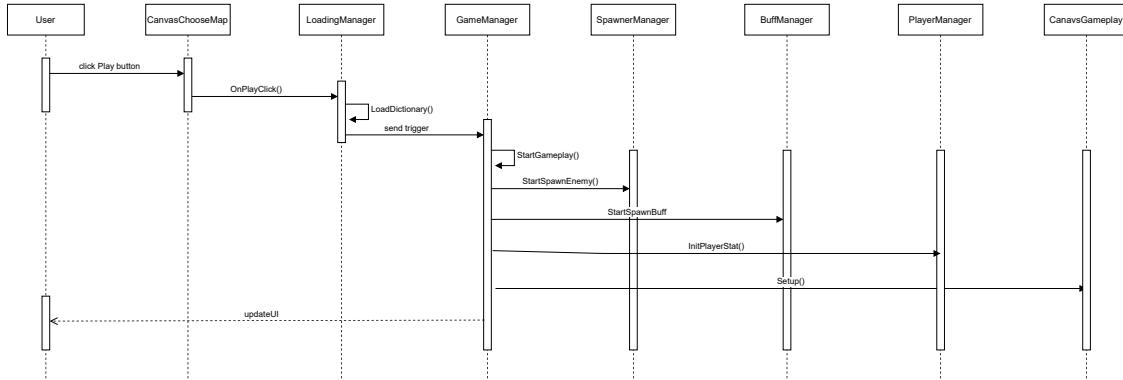


Figure 2.4: Sequence diagram of Start gameplay

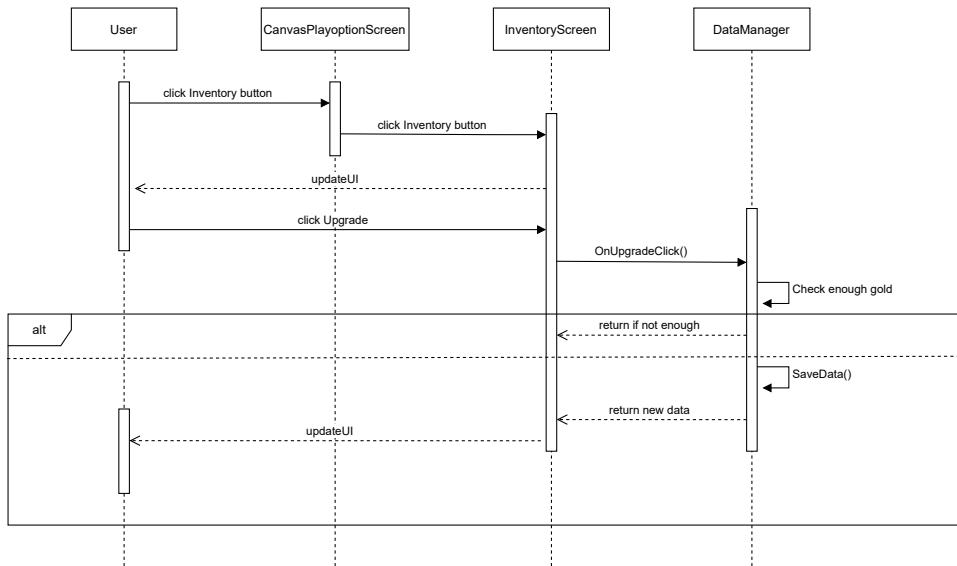


Figure 2.5: Sequence diagram of Upgrade

e, Game loop

Figure 2.6 is the sequence diagram of handle gameloop. After the player loses, the data will be updated and the main UI will switch to Canvas Choose Map.

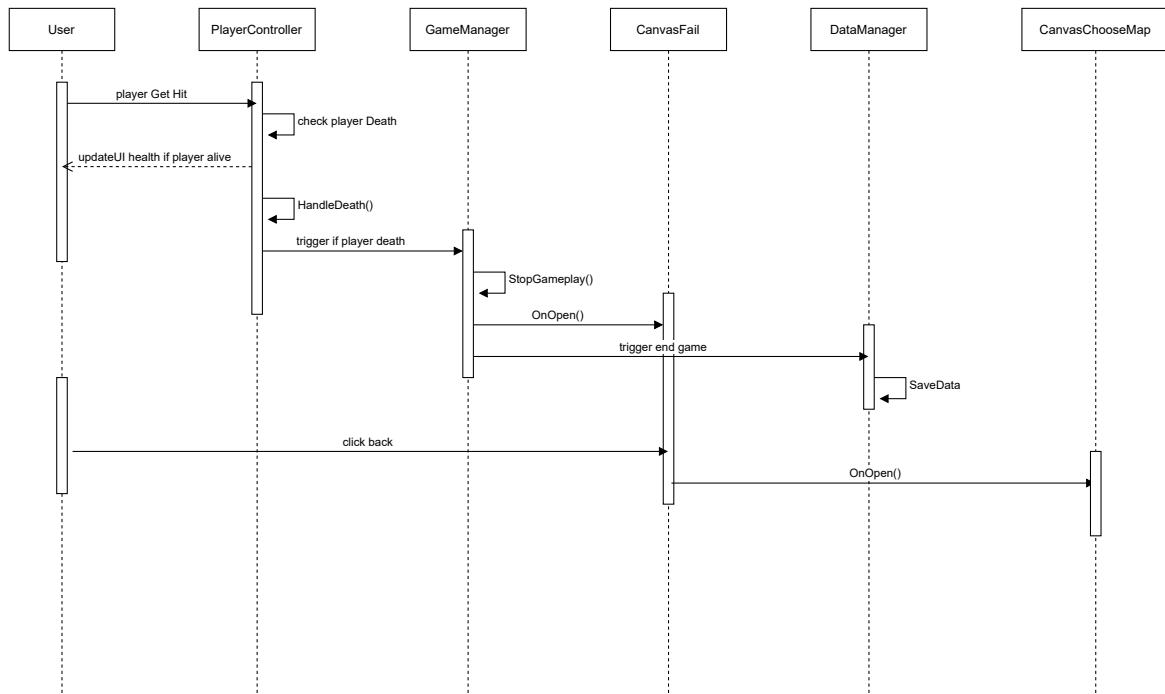


Figure 2.6: Sequence diagram of handle gameloop

2.2.3 Activity diagram

An activity diagram is a diagram that describes the execution steps, actions, decision nodes, and branching conditions to control the execution flow of the system. The following are the main sequence diagrams of the system.

a, Setting

Figure 2.7 is the activity diagram of Setting feature.

b, Highscore

Figure 2.8 is the activity diagram of Highscore feature.

c, Start

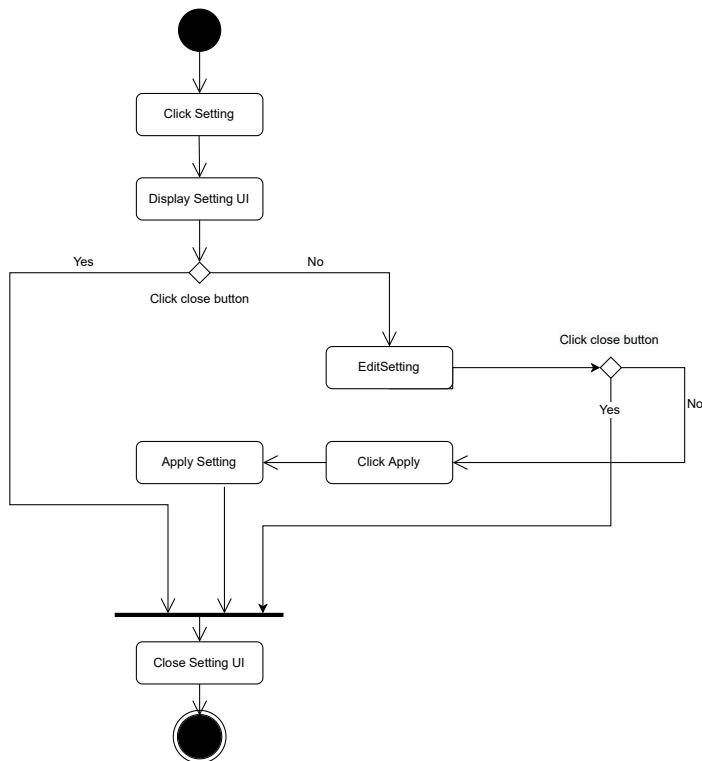
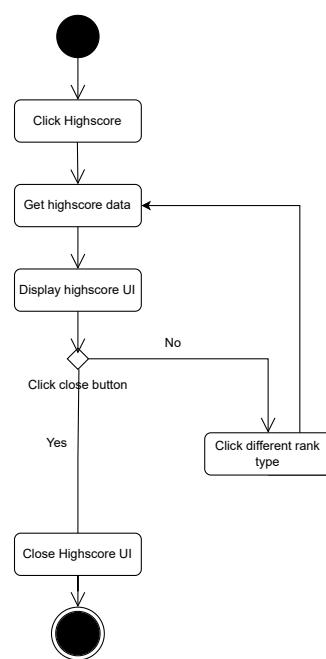
Figure 2.9 is the activity diagram of Start gameplay feature.

d, Upgrade

Figure 2.10 is the activity diagram of Upgrade feature.

e, Game loop

Figure 2.11 is the activity diagram of handle gameloop.

**Figure 2.7:** Activity diagram of Setting feature**Figure 2.8:** Activity diagram of Highscore feature

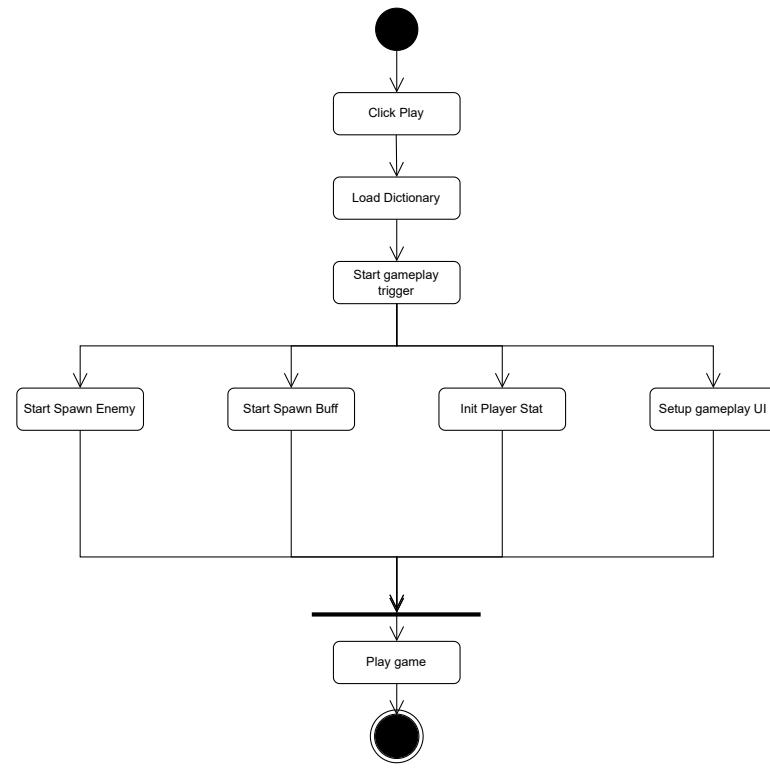


Figure 2.9: Activity diagram of Start gameplay feature

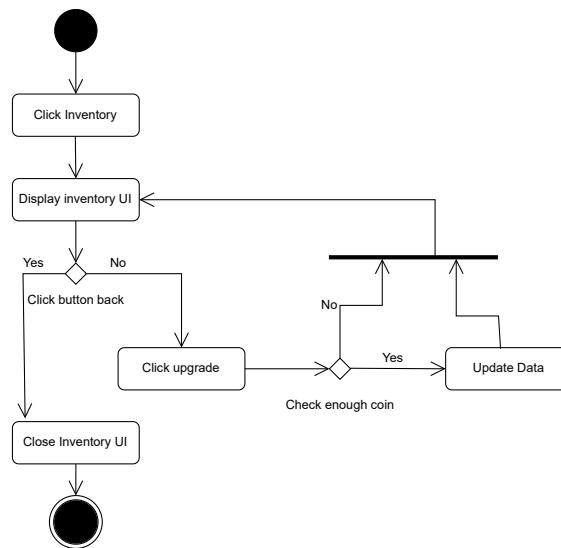


Figure 2.10: Activity diagram of Upgrade feature

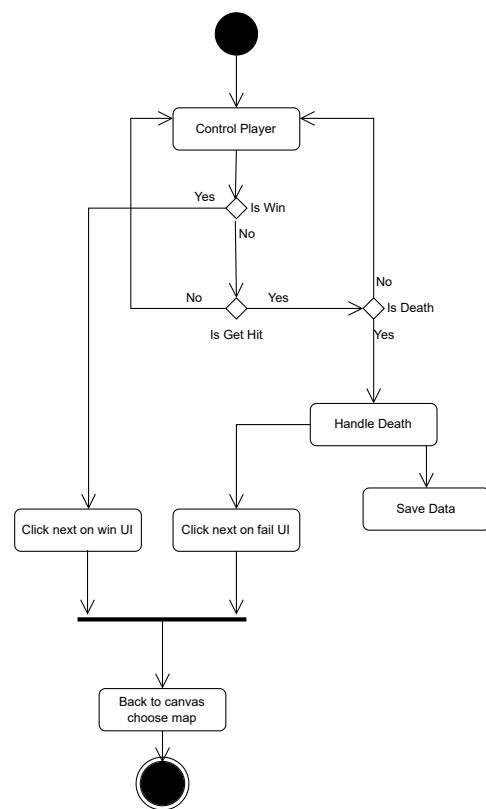


Figure 2.11: Activity diagram of handle gameloop feature

CHAPTER 3. GAME TECHNOLOGY AND DESIGN

3.1 Game technology

At the moment, due to the growing gaming industry, there is also a multitude of different development tools for game developers to choose from. Some typical names can be mentioned such as Unity Engine, Cocos Creator, Unreal Engine, Godot, etc. This project decided to choose Unity Engine as development tool. Unity has some great advantages such as free, large community, cross-platform, etc. Here is the basic information about Unity Engine.

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at the Apple Worldwide Developers Conference as a game engine. It supports Mac OS X. This tool has been gradually expanded to support a variety of desktops, mobile devices, consoles, and virtual reality platforms. It is especially popular for iOS and Android mobile game development and is considered easy to use for beginner developers and popular for indie game development. Unity gives users the ability to create games and experiences in both 2D and 3D, and it provides the main scripting API in C# using Mono, for both the Unity editor in plugin form and the game itself, as well as drag and drop functionality. Before C# was the main programming language used for the engine, it formerly supported Boo, the language was removed with the release of Unity 5 and a Boo-based implementation of JavaScript called UnityScript, was deprecated in August 2017, following the release of Unity 2017.1, in favor of C#.

Unity history: Unity 2.0 (2007) In 2007, Unity 2.0 was released with about 50 new features. The update came with a terrain engine that was geared for complex 3D scenes, as well as video playback, dynamic shadows that changed in real time, and other features. The update also included new tools that made it simpler for programmers to work together. It had a networking layer that offered Network Address Translation, State Synchronization, and Remote Procedure Calls, allowing programmers to create multiplayer games based on the User Datagram Protocol. Unity swiftly introduced support for the iPhone after Apple opened its App Store in 2008. The engine dominated the iPhone market for many years and gained popularity among iOS game creators.

Unity 3.0 (2010) In September 2010, Unity 3.0 was released, increasing the engine's graphical capabilities for desktop computers and gaming consoles. Along with Android support, Unity 3 included delayed rendering, a built-in tree editor, native font rendering, automatic UV mapping, audio filters, and the Beast

Lightmap tool from Illuminate Labs. VentureBeat noted in 2012, "Few businesses have helped the flow of independently developed games as much as Unity Technologies. Its tools are used by more than 1.3 million game creators to produce cutting-edge graphics for their iOS, Android, console, PC, and web-based games. Simply put, Unity wants to be the platform-agnostic game engine." According to a Game Developer magazine survey conducted in May 2012, Unity is the best game engine for mobile devices.

Unity 4.0 (2012) Unity 4.0 was released by Unity Technologies in November 2012. This version provided access to the Linux preview, support for DirectX 11 and Adobe Flash, as well as new animation tools dubbed Mecanim. In 2013, Facebook added a software development kit for games built using the Unity game engine. This included capabilities for tracking advertising efforts, deep linking-where users could jump straight from social media posts to particular game areas-and simple in-game image sharing. With Unity, Facebook created a brand-new PC gaming platform in 2016. Facebook's gaming platforms were supported by Unity, and Unity creators could export and publish games to Facebook more quickly.

Unity 5 (2015) The Verge said of 2015's Unity 5 release: "Unity started with the goal of making game development universally accessible. Unity 5 is a long-awaited step towards that future." With Unity 5, the engine improved its lighting and audio. Through WebGL, Unity developers could add their games to compatible Web browsers with no plug-ins required for players. Unity 5.0 offered real-time global illumination, light mapping previews, Unity Cloud, a new audio system, and the Nvidia PhysX 3.3 physics engine. The fifth generation of the Unity engine also introduced Cinematic Image Effects to help make Unity games look less generic. Unity 5.6 added new lighting and particle effects, updated the engine's overall performance, and added native support for Nintendo Switch, Facebook Gameroom, Google Daydream, and the Vulkan graphics API. It introduced a 4K video player capable of running 360-degree videos for virtual reality.

Some important concepts in unity:

Monobehaviour: MonoBehaviour is the base class that every Unity script derives from. It provides several lifecycle functions that make it easier to develop your apps and games.

GameObject: Unity defines GameObject as an object that represents the Assets in the game such as characters, plants, tools, props, cameras, effects... GameObject all contain a basic property called Transform, used to set GameObject's position, orientation and size.

Components: Components are properties added to GameObject such as Animation, sound, 3D models, effects, ... to build, combine different elements to define morphologies, behaviors,... of the desired object in the game.

Scenes: In Unity, a Scene is a game scene, the game space where the layout of GameObject is set, or a part containing interface settings such as in-game menus. Creating multiple Scenes will give flexibility in optimal resource allocation, managing segments in the game independently.

Scripts: Script are a Component in Unity. This is an essential component used to interact with player actions, or manage events to change the direction of the game in accordance with the game scenario. Unity gives programmers the ability to write script in C# language.

Prefabs: Prefab is a finished GameObject after adding properties and stored again for reuse. GameObject cloned from a prefab will be exactly the same. To create a prefab, simply drag a GameObject from the Hierarchy window onto the Project window

Camera: The camera in Unity is used to show the frame, the perspective that the player can see in the game. In a game, it is possible to set up multiple cameras to split the player screen, create effects, or customize such as view from the back, miniature map, ...

Navmesh: Navmesh is a specified grid in the Unity scene that specifies navigable areas in the environment, including areas where characters can walk, as well as obstacles. This is useful for situations where a combination of Pathfinding navigation and AI control is required. The Navmesh agent component helps characters avoid each other, move around the scene toward a common goal, or any other type of scenario involving spatial reasoning or navigation.

3.2 Game design

3.2.1 Core gameplay

lettThis project is a 3D top-down shooter built for Windows computers based on the context following the main character in the journey to atone for mistakes and save civilization for the world. The main character moves in the 3D coordinate system and hunts for monsters and survives until enough points pass the level. Player view from the top down, control the main character to move with the right mouse and shoot with the left mouse. The bullet of the main character fired will come with 1 letter, that letter will be selected by the player by pressing the character you want to shoot on the keyboard before firing. If the player shoots the letter correctly,

he will kill the monster and get extra points. A monster will represent a 1-word meaning in English but missing some characters, monster will be killed when the missing characters are shot enough at the person. The killed monster will display the Vietnamese meaning of the English word that the monster carries. On the map, there will be some points where beneficial buffs will appear for the player such as increased health, and increased speed, ... Over time when the player reaches the following levels, the difficulty of the vocabulary representing the monster will increase. Players can also upgrade the character's strength (speed, health, skills, ...) with the amount of gold accumulated when finish a level. Every time player wins a level, adding to the next UI will be opened, the system will save the time the player used to pass the level to create a high score table. The high score system will motivate the player to complete the level many more times. During the game, the player will be exposed to English words and see their meanings continuously, while also trying to remember and guess the hidden English characters. This will simultaneously use 2 methods of learning English, passive learning and active learning, which greatly increases learning efficiency.

3.2.2 Game loop

The Gameflow chart (Figure 3.1) visually represents the loop of stages players go through while experiencing the game. The gameflow of this project consists of 5 main states.

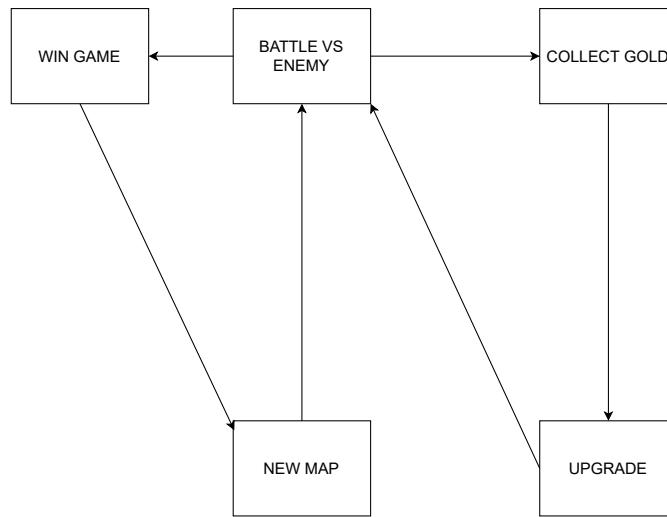


Figure 3.1: Player's skill 1

3.2.3 Game Unit

a, Player

Player (Figure 3.2) can move and shoot the selected character to destroy the monster, player can also use skills. Player will lose health if touched monsters and



Figure 3.2: Player

will die when out of health point. Player has 3 skills:

The first skill (Figure 3.3): Character will summon a sword and slash at the enemy in front, all hit monsters will show their hidden word.



Figure 3.3: Player's skill 1

The second skill (Figure 3.4): Character will power up and shoot spikes around, monsters hit will be stunned for a while and cancel the status of chasing the player.

The 3rd skill (Figure 3.5): Character will swing his shield in front of him, all monsthit monsters will be knocked back.

The 4th skill (Figure 3.6): Character dashes forward a short distance.

b, Monster

Monster (Figure 3.7) will patrol around and will rush at the player when it detects the player. Monster will die if get shot all hidden character or hit the player.



Figure 3.4: Player's skill 2



Figure 3.5: Player's skill 3



Figure 3.6: Player's skill 4



Figure 3.7: Monster

c, House

House (Figure 3.8) is the place where buffs are generated, the house can block the player's bullets. Neither monster nor the player can enter the house.



Figure 3.8: House

d, Well

The well (Figure 3.9) is a place where monsters spawn around, monsters and player cannot enter the well but player bullets can fly through the well.

e, Fence

Fences (Figure 3.10) block the path of player and monsters, bullets can fly through the fence.

f, Buff

Buffs (Figure 3.11) are appeared around the house, if the player touches a buff, there will be some beneficial effects such as increased health, increased movement speed,...



Figure 3.9: Well

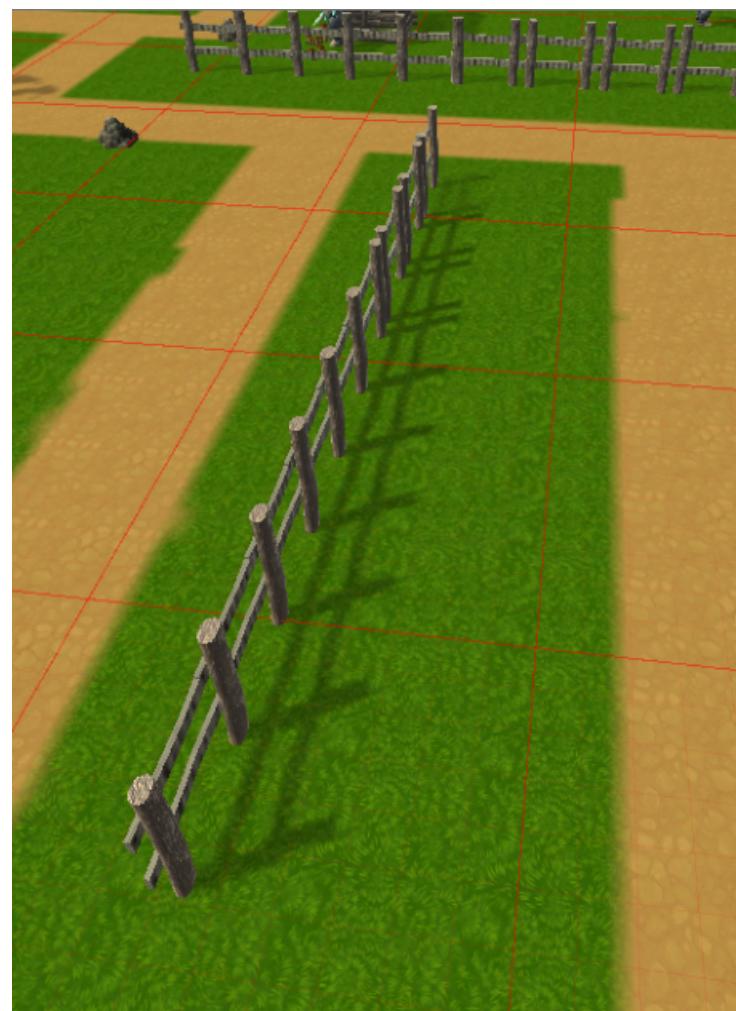


Figure 3.10: Fence



Figure 3.11: Buff

3.2.4 Enemy AI

The game's enemy is controlled by the state machine (Figure 3.12). When the monster spawns, it will be in the patrolling state, when it detects a player nearby (monsters will have a longer range to detect the player in front than the player behind) the monster will switch to the chasing state and increase its movement speed while chasing the player. If the monster is hit by the player's skill 2, it will be put into the stunning state, when the stun time expires, it will go to the patrolling state. If the monster is hit by the player's skill 3, its state will change to push back state, after being pushed back the monster will switch to the chasing state. When a monster is hit by all weak characters, it will enter the dying state and die. While in dying state monsters cannot be changed to another state.

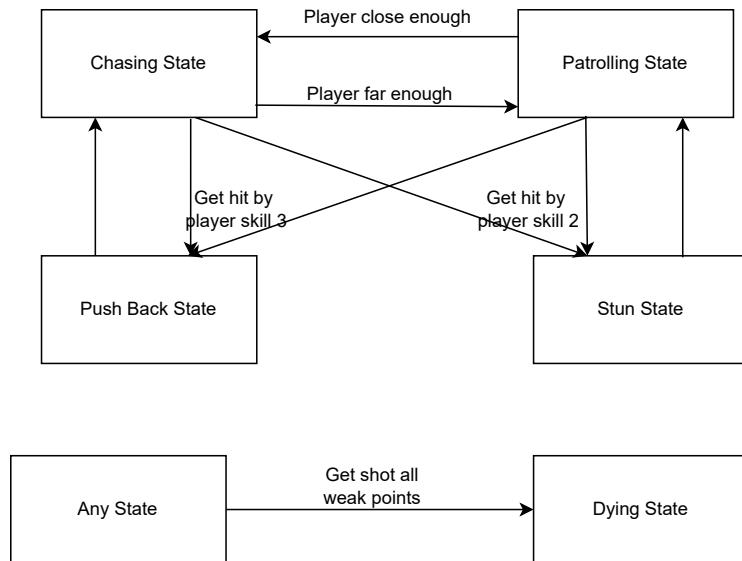


Figure 3.12: Enemy state machine

3.2.5 Vocabulary system

The Vietnamese meanings of words in this project have been refined to be as comprehensive as possible, helping players understand the most important meanings of an English word in Vietnamese. Monsters in each level will contain a random vocabulary in the list of words of that level. Before starting each level, the player can choose to see all the words that can appear in that level and learn it first. This makes it easier for players to win that level and also helps players learn more vocabulary. Once the monster has selected a word from the set, it will mask some random characters. The number of characters to be masked depends on the number of characters in that vocabulary, which will help the game be more balanced and also stimulate the player's vocabulary memorization.

3.2.6 UI Flow

The diagram 3.13 shows the transitions between the game's UI.

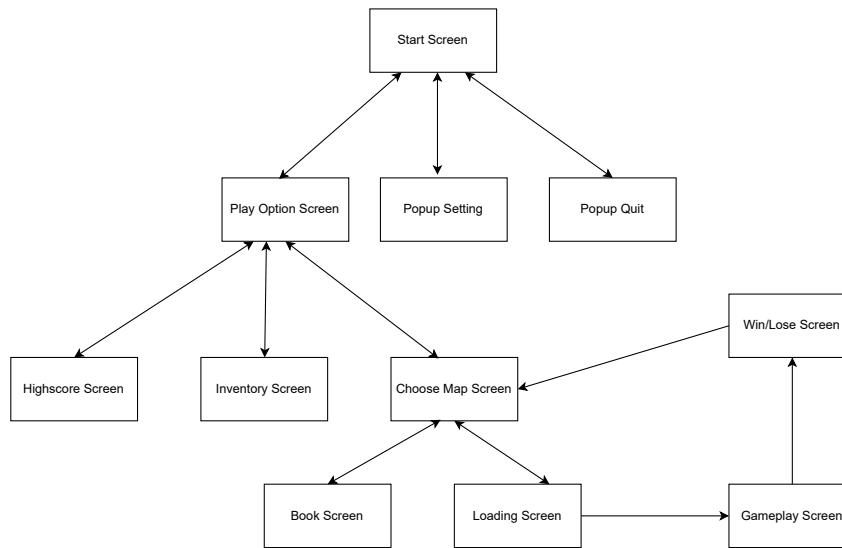


Figure 3.13: UI flow

3.2.7 A few main UI

a, Start screen

This is the first screen the player encounters when starting the game. The Start screen (Figure 3.14) has 4 options for the player: Quit, Play option screen, Setting and Tutorial screen.



Figure 3.14: Start screen

b, Highscore screen

This is the first screen players want to review the fastest winning times of each stage. The high score screen (Figure 3.15) will display the data of the selected player level, the player can choose back to return to the Play option screen.



Figure 3.15: Highscore screen

c, Inventory screen

The inventory screen (Figure 3.16) shows the amount of gold the player has and the power options the player can use the gold to upgrade. If the player presses Back, it will return to the Play option screen.

d, Choose map screen

The Choose map screen (Figure 3.17) is the level the player can choose to play. The project includes 5 gates with the names respectively A1, A2, B1, B2, and C1. The player can press Play to jump into the main gameplay or press Read to read the words the monsters at this level may contain. The stages the player cannot open will have a lock icon and in these stages, the player cannot interact.

e, Gameplay screen

The gameplay screen (Figure 3.18) is the UI when the player enters the gameplay. This screen includes the time from the start of the stage in the upper left corner of the screen, the minimap in the lower right corner, the money collected located in the upper right corner, the progress bar at the top, the skill and health bar at the bottom of the screen.

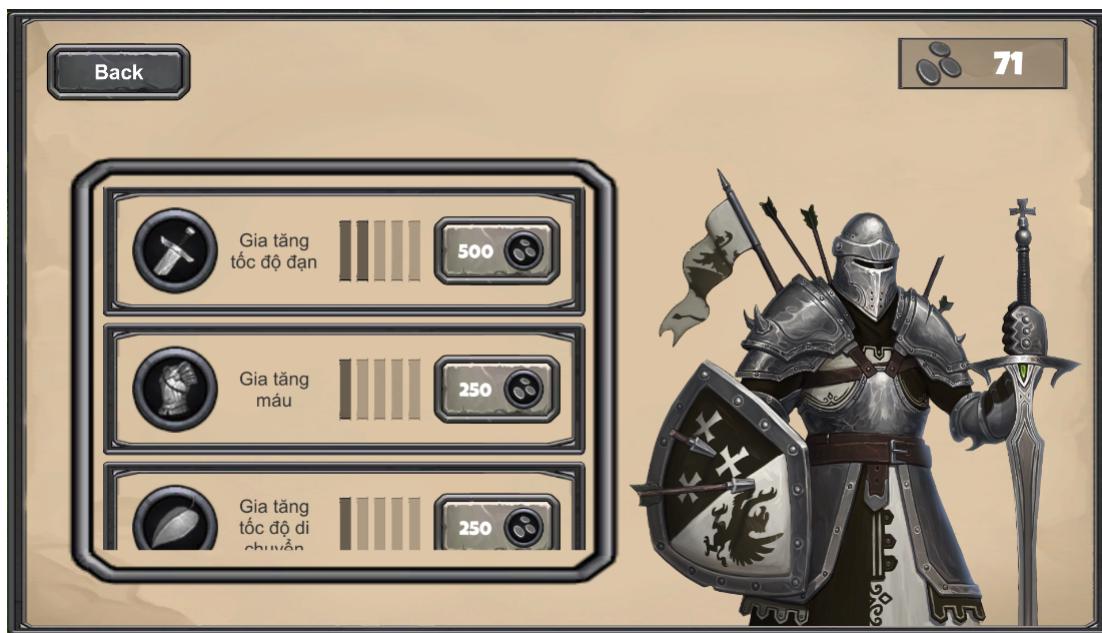


Figure 3.16: Inventory screen



Figure 3.17: Choose map screen



Figure 3.18: Gameplay screen

CHAPTER 4. EXPERIMENT AND EVALUATION

4.1 Test case

Software testing is the method of checking whether the software product conforms to the stated requirements, and ensuring that there are no bugs or defects. It includes the examination, analysis, observation and evaluation of various aspects of the product.

4.1.1 Control character movement

This test case is to test the player's ability when using the right mouse button for moving the character (Figure 4.1).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#1	Control character movement	The player has done all the operations to switch to the gameplay screen	1. The player uses the left mouse button to click on any point on the screen other than the UI	The character runs to the location where the mouse is clicked	Pass

Figure 4.1: Test case for control character movement

4.1.2 Control character shoot

This test case is to test the player's ability when using the left mouse button for controlling the character to shoot (Figure 4.2).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#2	Control character shoot	The player has done all the operations to switch to the gameplay screen	1. The player uses the right mouse button to click on any point on the screen other than the UI	The character shoots to the location where the mouse is clicked	Pass

Figure 4.2: Test case for control character shoot

4.1.3 Cast character skill 1

This test case is to check the character's ability to use the first skill (Figure 4.3).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#3	Check character skill 1	The player has done all the operations to switch to the gameplay screen	1. control the character to move to face the monster 2. Press button 1 on keyboard	The character summons the sword slashes down forward, all hit monsters show their hidden word	Pass

Figure 4.3: Test case for cast character skill 1

4.1.4 Cast character skill 2

This test case is to check the character's ability to use the second skill (Figure 4.4).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#4	Check character skill 2	The player has done all the operations to switch to the gameplay screen	1. control the character to move to face the monster 2. Press button 2 on keyboard	The character casting sword pokes from the ground up around the place where he is standing, all hit monsters get stun	Pass

Figure 4.4: Test case for cast character skill 2

4.1.5 Cast character skill 3

This test case is to check the character's ability to use the third skill (Figure 4.5).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#5	Check character skill 3	The player has done all the operations to switch to the gameplay screen	1. control the character to move to face the monster 2. Press button 3 on keyboard	The character swings his shield forward, all hit monsters are pushed back	Pass

Figure 4.5: Test case for cast character skill 3

4.1.6 Cast character skill 4

This test case is to check the character's ability to use the fourth skill (Figure 4.6).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#6	Check character skill 4	The player has done all the operations to switch to the gameplay screen	1. Press space on keyboard	The character flashes forward a short distance	Pass

Figure 4.6: Test case for cast character skill 4

4.1.7 Check character and enemy collision

This test case is to check the collision between the character and the enemy (Figure 4.7).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#7	Check character and enemy collision	The player has done all the operations to switch to the gameplay screen	1. Control the player to collide with monster	The character has health <u>subtracted</u> , the monster dies	Pass

Figure 4.7: Test case for check character and enemy collision

4.1.8 Check character die

This test case is to check whether the character dies when runs out of health point (Figure 4.8).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#8	Check character die	The player has only less than 2 hp left	1. Control the player to collide with monster	The character dies, the monster dies. Lose UI appear	Pass

Figure 4.8: Test case for check character die

4.1.9 Check enemy and bullet collision

This test case is to check the collision between the character and the enemy (Figure 4.9).

4.1.10 Check upgrade system

This test case is to check the game's upgrade system (Figure 4.10).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#9	Check bullet and enemy collision	The player has done all the operations to switch to the gameplay screen and the map has monster	1.Shoot the bullet toward monster	If the monster is hit with the right char, the hidden text will appear. If all the char appear, the monster will die. The bullet that hit the monster will disappear	Pass

Figure 4.9: Test case for check enemy and bullet collision

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#10	Check upgrade system	The player access to Inventory UI	1.Click buy on the stat player want to upgrade	If money is enough, do upgrade and update UI. Player get higher stat in the next gameplay	Pass

Figure 4.10: Test case for check upgrade system

4.1.11 Check high score system

This test case is to check the game's high score system (Figure 4.11).

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#11	Check high score system		1.Play game play and win a few times 2.Acces high score UI	The time required to win each time is saved and sorted from low to high	Pass

Figure 4.11: Test case for check high score system

4.1.12 Check setting system

This test case is to check the game's setting system (Figure 4.12).

4.1.13 Check vocabulary system

This test case is to check the game's vocabulary system (Figure 4.13).

4.2 Evaluation

Basically, this project has met the basic requirements set out from the beginning.

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#12	Check setting system		1.Click setting 2.Edit sound config and apply	Sound volume is adjusted as in setting	Pass

Figure 4.12: Test case for check setting system

Test case Id	Test case name	Pre-condition	Step	Expected Respond	Pass/Fail
#13	Check vocabulary system	The player has done all the operations to switch to the gameplay screen	1.Control the character move around map and kill monster	All monsters belong to the current level's vocabulary. The number of hidden characters matches the total number of characters. The Vietnamese meaning of the words matches the English vocabulary	Pass

Figure 4.13: Test case for check vocabulary system

Test cases are executed and there are no errors. The performance of the game is also good and stable. From the start, the application takes up RAM that ranges from 370 megabytes to 400 megabytes and about 12% - 16% of the CPU usage from a mid-range computer. The amount of resources occupied does not change from the beginning to the end. In conclusion, the game does not require high computer configuration and can be played on most current computer lines on the market.

CHAPTER 5. CONCLUSION

5.1 Solution and Contribution

5.1.1 Game graphic and sound

a, Problem

To increase the game feel for players, the game's graphics and sound are paramount. The selection of assets in this project encountered some difficulties. The first is that although there are a lot of resources online, it is also very time-consuming to search because you have to try many different asset sets. Next, due to the target audience of the game, the game's graphics are not required to be too heavy and run smoothly on popular computers. In addition, the assets between the model, UI, and sound must have a good combination to increase the feeling of the player.

b, Solution

To choose the right asset, the first thing we need to do is to build a plot for the game. Then all the resources to find will be based on that topic. A few key keywords will be selected and attached to each search order for all different asset types. The next thing is to build a system that must be flexible and modular to make it easy to change assets but still keep the game flow running as it was. This requires some knowledge of object classification and design patterns. In the process of making, small details also need to be paid attention and meticulously completed because this is a very important factor to increase the "game feel" for the player. The last thing is to constantly search and change assets until a satisfactory set of assets is found.

5.1.2 Scalability

a, Problem

Since there are many ideas yet to be implemented, the system needs to be able to be easily modified and developed. The initial version of the system may run smoothly, but further development will result in very large code files and overlapping game logic.

b, Solution

Several designs have been made to keep the system open. The first change is to use a state machine for the enemy, the state machine will help to split a large code file into small independent code files that are easy to track and expand. It also helps to slightly improve performance when not having to check too many conditions in 1 frame. Next is the rational use of Singleton. Singleton is a very

efficient way to access data between classes, but as the system gets bigger, having too many singleton classes will cause the logic to be linked together and lead to changing one logic file which will affect to other files.

5.1.3 Performance

Game performance is how well a computer can handle and display games and is one of the crucial elements of a game. This project has had good performance when running, but in order to ensure that computers with weaker configurations achieve good performance when executing the game, some changes and solutions need to be made.

a, Problem

Since there are many ideas yet to be implemented, the system needs to be able to be easily modified and developed. The initial version of the system may run smoothly, but further development will result in very large code files and overlapping game logic.

b, Solution

In Unity there are 2 main reasons that can affect game performance, maybe due to errors in the code or because the graphics are too heavy for the processing capabilities of the machine. To improve game performance, the first thing the project does is implement a pooling system. Because the Instantiate() and Destroy() functions are quite heavy when done in large numbers, it will easily cause lag in weakly configured computers. There are two groups of objects in the game that need to be destroyed and initialized continuously, Enemy and Bullet, so these two groups of objects have been applied with a pooling system to improve performance. Caching is also used a lot to reduce the frequency of using Unity's get component functions and C#'s garbage collections.

The game's rendering capabilities need to be improved next. The system has implemented a few strategies to decrease the quantity of draw calls sent to the GPU in order to solve this issue. A protracted frame and fps loss will result from an excessive amount of draw calls because each one requires CPU processing time to gather information, so when the number of draw calls is too large, it will lead to a prolonged frame and fps drop. The game used the static batching method - this method will make all meshes that share the same material (shaders, textures, ...) will be able to be pooled into one draw call.

5.2 Conclusion and Future Work

This project is a Top-down action game with purpose to help players learn English while playing. The game has completed essential functions such as controlling the player and using the player's skills to interact with the enemy, deploying an AI system for the enemy based on the State machine, combining logic and animation for the player and enemy to bring smooth and realistic interaction for players, build data saving and loading system. From this system, the project has implemented other functions such as the upgrade system for the player and the high score system for the player to help the player save the times of all fastest wins. In addition to optimizing game performance, this project has integrated a pooling system for spawning monsters and bullets. All user interface (UI) such as Start, Inventory, Rank, etc. have also been completed. However, this project still has some shortcomings. The first is that the game's graphics and sound system are not good enough to bring emotions to the player. The reason for this is the lack of suitable assets with affordable price. Next, due to the limited game design experience, the UI layout is still basic and lacking in creativity, the map in the game is small, so it is easy to be boring for players. Finally, some initial ideas have not been implemented in the project yet such as custom game mode and difficulty adjustment feature of the game.

There are a few features of the project that I plan to implement in the future. The first is to add the difficulty of the game, I plan to add other states to the enemy's state machine so that when the player adjusts the difficulty of the game, in addition to changing the parameters, the enemy will also appear other behaviors corresponds to the difficulty the player chooses. For example, when at a higher difficulty level, the monster may have some additional abilities such as stealth and a more complex movement trajectory to dodge the player's bullets, Next is the custom mode in the game, this mode will help players create their word boards and when starting the gameplay, the monsters will contain words that are the words that the player has created. This mode will help players learn the words they want. Another feature I want to add is to create more maps or the program will have an automatic map creation feature, aiming to prevent players from getting bored when playing the same map over and over again. Finally, it is to improve the graphics and sound of the game to increase the emotions of the players.

During the project, I also learned many things. Starting from recognizing the problems when the scale of the project increases, there have a deeper understanding and experience in choosing the design pattern that best suits each part of the project. Next is understanding the importance of clean code as well as applying

appropriate data structures and algorithms to optimize performance. I also have more experience in handling the animation and particles in unity in the process of making player skills. Finally, I would like to once again thank my teachers, family and friends for always helping me during the completion of the project, I hope my product will be well received and have positive contributions for society.