



AHMED MUSTAFA
01-134182-073

Wisteria

Bachelor of Science in Computer Science

Supervisor: Ms. Erum Ashraf

Department of Computer Science
Bahria University, Islamabad

May, 2022

Certificate

We accept the work contained in the report titled “**Wisteria**”, written by AHMED MUSTAFA as a confirmation to the required standard for the partial fulfilment of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Ms. Erum Ashraf

Internal Examiner:

External Examiner:

Project Coordinator: Dr. Moazam Ali

Head of the Department: Dr. Arif Ur Rahman

May 2022

Acknowledgments

To start off we would like to thank our supervisor Ma'am Erum Ashraf. We acknowledge Ma'am Erum Ashraf's remarkable guidance regarding academic as well as administrative procedures. Being the supervisor for this project, she managed to provide valuable feedback and suggestions on the project and its documentation. She was always there as a mentor whenever her guidance was needed. We are grateful to Ma'am Erum Ashraf for her cooperation and support in the evaluation of our research and for providing us with her valuable feedback throughout the research work. We would like to thank our family who made all this project possible. It could not be possible without their support in any regard. Our sincere thanks to the Head of the Computer Science Department at Bahria University Islamabad for providing us with a great opportunity to conduct our research at the institution and also for providing us access to the resources and research of this institution.

AHMED MUSTAFA

ISLAMABAD, PAKISTAN

MAY, 2022

*“I feel that the most important requirement in success is learning to overcome failure.
You must learn to tolerate it, but never accept it. ”*

Reggie Jackson

Abstract

Wisteria is a game designed to show that as much as graphics and storyline are important in a game, so is the logic of the game AI. Improving the gaming experience of the players has been the core focus of the research being conducted on game development. Improving the game AI has been a contemporary challenge in game development. Many modern games have better storylines and improved graphics but are not liked by the users because they have ordinary and lackluster AIs. Users play the games because of the interesting stories and fancy graphics but soon lose interest because of the subpar AIs. Many studies have found that improving the game AI is important for keeping users interested in games. Unlike many other games released in the market, this project is designed to cater to users that are used to playing similar games and also that are new to games as well. By using minimal AI features this project contains an AI that does not have much complex and unnecessary functionality. The game AI has been created using a behavior tree that is used for making the AI follow advanced patterns according to the situation it will face. The game also includes modern video game features such as loot bags, leveling, dynamic animations, an open world map, and character selection so as not to bore the player and keep them occupied.

Contents

Acknowledgments	i
Abstract	iii
Abbreviations	x
1 Introduction	1
1.1 Introduction	1
1.2 Objective	1
1.3 Problem Description	2
1.4 Story Line	2
1.5 Research	2
1.6 Project Scope	3
1.7 Methodology	3
1.8 Feasibility Study	4
1.9 Solution Application Areas	4
1.10 Tools/Technology	4
2 Literature Review	5
2.1 Introduction of the Gaming Industry	5
2.2 The Future of the Gaming Industry	5
2.3 Advancement of AI in Games	6
2.4 Examples of Games with a Good AI Implementation	6
2.4.1 Civilization series	6
2.4.2 Rocket League	7
2.4.3 Tom Clancy’s Splinter Cell: Blacklist	7
2.4.4 F.E.A.R	8
2.4.5 Alien Isolation	9
2.4.6 Octopath Traveler	10
2.5 Wisteria	10
3 Software Specifications	11
3.1 Proposed System/Game	11
3.2 Functional Requirement	12
3.2.1 Functional Requirement	12
3.2.2 Non Functional Requirements	13
3.3 Use Cases	14

4	System Design	28
4.1	System Architecture	28
4.1.1	Presentation Layer	28
4.1.2	Logic Layer	28
4.1.3	Database	28
4.1.4	Design Constraints	29
4.2	Design Methodology	29
4.3	High Level Design	29
4.4	Low-Level Design	30
4.5	A.I Implementation	32
4.6	Algorithms used	33
4.6.1	The A* Algorithm	34
4.7	GUI Design	34
5	System Implementation	37
5.1	System Architecture	37
5.1.1	The Video Game	37
5.2	System Internal Component	37
5.2.1	Health	38
5.2.2	Mana	38
5.2.3	Spell Casting	38
5.2.4	Regeneration	38
5.2.5	Item Pickup	38
5.2.6	Ability cooldown	39
5.2.7	Loot/Bag pickup item	39
5.2.8	Melee combat system	39
5.2.9	Character movements and animations	39
5.2.10	Character selection	39
5.2.11	Quest System	39
5.2.12	Player levelling	39
5.2.13	Inventory	40
5.2.14	Shops	40
5.3	Tools and Technology	40
5.3.1	Unreal Engine	40
5.3.2	Microsoft Direct X	40
5.3.3	C++	40
5.4	Methodology	40
6	System Testing and Evaluation	42
6.1	Introduction	42
6.2	Graphical User Interface Testing	42
6.2.1	Integration Testing	43
6.3	Exception Testing	45
6.4	Performance Testing	46
6.5	Exception Testing	48
6.6	Performance Testing	49

7	Conclusions	51
7.1	Conclusions	51
7.2	Future Enhancements	51
A	User Manual	52
A.1	Introduction	52
A.2	System Requirments	52
A.3	Installation	52
A.4	Settings	53
A.5	Controls	53

List of Figures

2.1	Civilization.	6
2.2	Rocket League.	7
2.3	Tom Clancy’s Splinter Cell Blacklist	8
2.4	F.E.A.R	9
2.5	Alien Isolation	9
2.6	Octopath Traveler	10
3.1	Flow of proposed system	12
3.2	Use Case of Game Flow	14
4.1	Agile Methodology Life Cycle	29
4.2	Block Diagram	30
4.3	Activity Diagram	31
4.4	Sequence Diagram	32
4.5	RPG HUD Display	34
4.6	In Game View	34
4.7	In Game View with Roaming AI	35
4.8	Class Select Screen	35
4.9	Melee Character In Game View	35
4.10	In Game View with Inventory Accessed	36
4.11	In Game View with Item in Inventory after Item Pickup	36
4.12	Loot Bag Screen	36

List of Tables

3.1	Flow of Game	15
3.2	Main Menu Use Case	16
3.3	In-game Interface widget	17
3.4	Health Functionality	18
3.5	Mana Functionality	19
3.6	Spell Casting Functionality	20
3.7	Health and mana regeneration.	21
3.8	Health pickup	21
3.9	Ability cooldown	22
3.10	Character movement and animation	22
3.11	Character selection	23
3.12	Quest system	23
3.13	Player leveling system	24
3.14	Inventory system	25
3.15	Loot bag system	26
3.16	Shop system	27
6.1	Main Menu UI Test Case	43
6.2	In-Game UI Test Case	43
6.3	Health Functionality Test Case	43
6.4	Mana Function Test Case	44
6.5	Spell Casting Test Case	44
6.6	Health and Mana Regeneration Function Test Case	44
6.7	Health and Mana Pickups Test Case	45
6.8	Ability Cooldown Test Case	45
6.9	Character movement and animation Test Case	45
6.10	In-Game Functionality Exception Test Case	46
6.11	Character Movement and Physics Test Case	46
6.12	Overall Game Performance Test Case	46
6.13	Character Selection Test Case	47
6.14	Quest System Test Case	47
6.15	Player Levelling Test Case	47
6.16	Player Inventory Test Case	48
6.17	Shop Test Case	48
6.18	Main Menu Exception Test Case	48
6.19	In-Game Spell Exception Test Case	49
6.20	Character Movement and Physics Test Case	49

6.21 Overall Game Performance Test Case 50

Acronyms and Abbreviations

AI	Artificial Intelligence
RPG	Role Playing Game
GB	Giga Byte
3D	Three Dimensional
2D	Two Dimensional
J R R Tolkein	John Ronald Reuel Tolkien
GHz	Giga Hertz
VR	Virtual Reality
AR	Augmented Reality
UI	User Interface
GUI	Graphical User Interface
HUD	Heads-Up Display
SDK	Software Development Kit
OS	Operating System

Chapter 1

Introduction

1.1 Introduction

The project idea is to create a third-person open-world role-playing game (RPG) consisting of enemy A.I (the enemy that will fight with the player) will interact with the player in a 3D environment. The game will consist of modern video game elements like inventory management, loot bags that are obtained from defeating the enemy that will reward the player with items and the gold coins that are the game's currency, a quest system to obtain missions to complete, a shop where the player can use the currency to exchange for items and many other modern elements that exist in modern games today.

1.2 Objective

- To develop a 3D-Open World Role Playing Game inspired by the fantasy novel Lord of the Rings by J.R.R. Tolkien [1].
- To add a game A.I. for player immersion that will interact with the player when entering the sight of the A.I.
- To create a game that the player will enjoy without the meaningless elements that exist in modern games.
- To see the true limitations of A.I in games.

1.3 Problem Description

To create a game for entertainment purposes, the ongoing pandemic has caused a lot of distress in the people so to counter this problem my game will provide an immersive experience without many of the elements that exist in modern games that add a difficult learning curve for a player that a new to gaming and old. The current games that are developed these days have more of a focus on storyline, characters, and graphics instead of on the game A.I itself, which leaves room to improve the A.I, Which in terms will overall improve the experience of the player providing them with a more realistic experience.

1.4 Story Line

The story of most action games is simple and easy for the user to understand without being too complex or generic for the player to get bored of too easily. The story will begin in a village in an unknown kingdom where the main character named by the player, will have his village attacked by a corrupt lord of the kingdom whose main purpose is to cause a disparity between the two kingdoms the current kingdom of the player and another neighboring kingdom to plunge them into war, so as to gain the favor of an evil faction who wish to sacrifice a large number of the population to summon a powerful demon to gain power and wealth. The player's gameplay will be centered around stopping this from happening and getting revenge for the murder of his entire village.

The gameplay will consist of different quests such as either fetching some item from someplace, enemy extermination, and helping different NPCs with different quests that the player can take.

1.5 Research

Improving the gaming experience of the players has been the core focus of the research being conducted on game development. Improving the AI has been a contemporary challenge in game development. Many modern games have better storylines and improved graphics but are not liked by the users because they have ordinary and lack luster AIs. Users play the games because of the interesting stories and fancy graphics but soon lose interest because of the subpar AIs. Many studies have found that improving the AI is important for keeping users interested in the games. There are many hurdles in improving ordinary AIs. A study has investigated the difficulties faced while reforming AI (NPCs) in games. They found three outstanding challenges; first, the developers have limited time to develop the game AI, and they focus more on other aspects of the game like graphics, storyline, and

network connections. Second, despite the fact that AI controls the NPCs in the game still, the development of environments (level design) is typically done independently of the development of the game AI. Third, the games are constantly being updated to improve the gaming experience of the players this poses significant challenges for improving the game AI [2].

Another challenge faced by the developers in improving the AIs of the games is due to the advanced graphic fidelity of modern video games and the application of technologies like Virtual Reality and Augmented Reality, due to this, players have been enjoying the immersion and playability of games now more than ever, but it poses a great challenge on the developers and designers to create more intelligent NPCs to build more engaging and challenging games [1].

Player experience modeling, as a major branch in player modeling, aims to model how the player feels during game-playing. As mentioned earlier, the main motivation of game AI is to provide players with a more immersive and interesting game experience, so the model of player experience is also of importance, yet it is not being focused more [1].

1.6 Project Scope

The gaming platform is overwhelmingly vast. There are games for each different individual and cater to how they want to enjoy these games. However many games today have a difficult learning curve and unnecessary elements that can stress the user that is either familiar with how these games work or have just started to get into gaming. This project unlike others does not contain many of the complex functionalities that can frustrate the user.

The game does not have an age restriction and can be played by almost anyone as it is easy to understand and does not contain unnecessary elements. The game will contain a simple A.I will have the necessary features to not make it lacking in respect to other games.

1.7 Methodology

This project will greatly benefit from a tried and trusted approach. The agile approach manages a project by dividing a huge project into smaller chunks of attainable goals. It includes a collaboration between stakeholders and developers to complete the project according to the requirements within the specified time. Every phase follows an incremental approach that follows a development cycle consisting of planning, executing, and evaluating [3].

1.8 Feasibility Study

1. Risks Involved

There is a possibility that the game does not fulfill the expectation of the fantasy genre fan base which may result in backlash and negative feedback.

The content of the game includes flashing screens which may cause seizures.

2. Resource Requirement

- Desktop PC or Mac.
- Windows 7 64-bit or Mac OS X 10.9. 2 or later.
- Quad-core Intel or AMD processor, 2.5 GHz or faster.
- NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series card or higher.
- 8 GB RAM [4].

1.9 Solution Application Areas

If the project becomes successful, additional features can be added which will greatly improve the experience provided. The project will be built using the free assets that are available for use by the public. The game will feature on steam and Epic games stores which will allow the masses to not only play and experience and provide their feedback on the project.

1.10 Tools/Technology

- Unreal Engine.
- C++.
- Microsoft Direct X.
- Vulkan.
- Adobe Photoshop / Lightroom.
- Adobe Premier / After effects.

Chapter 2

Literature Review

2.1 Introduction of the Gaming Industry

The gaming industry is no longer restricted to a certain age group or consumer segment, With the advent of mobile gaming and improvements to hardware used in playing these games, gaming has become a viable form of entertainment for players from all backgrounds and ages. This switch to the mainstream has also meant an increase in revenues generated by the industry with about US \$180.3 billion in 2021 alone with an increase of 1.4% over the last year of 2020, and 2020 showed a remarkable increase of 12% than the previous year of 2019. The improvements to hardware such as sound cards, graphics, and faster processors have meant a related growth and development of the gaming industry as well. As a result, modern games, especially those that are PC based, have become very demanding as applications and serious gamers are among those who purchase high-powered personal computers to keep up with the newest games.

2.2 The Future of the Gaming Industry

The gaming industry is an entertainment industry that is continuing to grow at a rapid rate and shows no signs of stopping. Many industries are now reliant on gaming and have shown remarkable improvements such as the improvement of the graphics card, gaming engines, and even the music industry, as having a rich story with amazing and stunning graphics paired together with music that captivates the hearts and minds of the player has led to many industries heavily reliant on the gaming industry to survive.

Now the gaming industry has led to the birth of new technologies such as the introduction of Virtual Reality (VR) and as well as Augmented Reality (AR) into gaming which has led to its rise and will continue to do so.

2.3 Advancement of AI in Games

Artificial Intelligence, is the emulation of human intelligence in machines. And, in the case of game development, its underlying objective is to make video games intelligent. Speaking of the specifics, it provides the games with the ecosystem that supports intelligent game behaviors for non-player characters and provides smart game control to make the interaction and movement between the game character smarter, and close to reality.

In the coming times, it is predicted that the benefits of Artificial Intelligence on games will continue to grow and will be more positive than not, making games more immersive, realistic, and life-like. Intelligent game behavior will provide attractive features in terms of realistic movement and interactions between game characters and game players.

2.4 Examples of Games with a Good AI Implementation

There are many games present that have a good implementation of AI and have shown what can be achieved if the proper implementation is carried out.

2.4.1 Civilization series

In the civilization series, the player is tasked with choosing a civilization and building their own empire while battling for supremacy against either other human players or the game's AI. The player can achieve victory through either total domination by defeating the other AI or through using diplomacy within the game to create allies and using different strategies to achieve victory. However, achieving victory is not easy as the game's AI also has its own strategies and makes sure to either give the player a hard time or totally defeat the player by using different strategies to achieve its goals.



Figure 2.1: Civilization.

2.4.2 Rocket League

Rocket League is a high-powered hybrid of arcade-style soccer paired vehicles that you can customize with either players or the game's AI. The game offers a totally different playstyle as you control different cars and must guide the large soccer ball to the opposite side's goal. The game is a classic twist on the traditional game of soccer. However, to achieve a feeling of competitiveness the game must have a strong AI to give the player a challenge and that is what the game has managed to provide. Although it looks like any sports video game, the game AI is very powerful to handle the ball game strategy, which in turn provides a realistic experience.

RLBot is a community of programmers who work to create high-level Rocket League bots using Artificial Intelligence. Some of these AI bots possess an amazing level of mechanical skills, they're able to hit advanced aerial shots, nasty redirects, and even incredibly smooth flicks totally autonomously without any input from players [5].



Figure 2.2: Rocket League.

2.4.3 Tom Clancy's Splinter Cell: Blacklist

Tom Clancy's Splinter Cell: Blacklist is an action-adventure stealth video game where the player can choose to either play the game stealthily or be aggressive in their gameplay. During Blacklist's development, its team faced a variety of challenges. The first was to create stealth which was satisfying and fun for players. According to the team, players gain satisfaction from stealth with freedom and choices which allow them to develop a plan. Players must experiment, with each decision having consequences. Game director Patrick Redding compared it to the development of an ecosystem; the team designed a dynamic

artificial intelligence that would react differently to players' actions, making levels feel alive and adding randomness. Encouraging the "panther" style of play (aggressive stealth), the team incorporated elements from the original Tom Clancy's *Splinter Cell* (where one mistake would abort a mission) and *Conviction* (where stealth seamlessly becomes combat). This approach, the team thought, could help players to feel like elite, silent predators [6].



Figure 2.3: Tom Clancy's *Splinter Cell Blacklist*

2.4.4 F.E.A.R

F.E.A.R or First Encounter Assault Recon is a survival horror first-person shooter released in 2005, the game follows the players who are a part of a special force (F.E.A.R) sent to take control of a research institution that is creating super soldiers.

The games even though were launched in the year 2005 is still believed to be having the best A.I in first-person shooter games to date. The games A.I is rather simple even though it looks complex. The team which was responsible for the game's development had only one A.I programmer, and so designing a 'planning'-based system that lets the AI soldiers think for themselves. Instead of telling an AI exactly how to behave in every situation, F.E.A.R gives the soldiers a set of goals and a set of possible actions and then lets them figure it out for themselves. Essentially, the AI reacts dynamically to what's going on in the environment – for example, if they're in danger they will look to retreat, but only if they can identify a safe path to follow. If not, they will hunker down, and perhaps blind fire from cover to try and slow your progress [7].



Figure 2.4: F.E.A.R

2.4.5 Alien Isolation

Alien isolation is a recent game that is one of the best games in how it uses the A.I and level design in the game to scare the player and to give the player the impression that the A.I is learning from the player themselves. However, this is not true, the game does have two A.I agents, the director, and the alien itself. The game is designed in such a way that the director A.I sends the alien or the hunter A.I to scare the player and give a sense of dread. However, the game also has some leniency so that the player does not consider the game to be outright cheating or too hard to either play or enjoy [8].



Figure 2.5: Alien Isolation

2.4.6 Octopath Traveler

Octopath Traveler is a beautiful game that uses 2D aspects with a 3D environment with colorful graphics that draws the player, paired with its open world and 8 different story paths of each individual character that draw the player into the story and the world. However, what the game lacks is improvement in its combat or the A.I.



Figure 2.6: Octopath Traveler

2.5 Wisteria

The game being created is being designed in the Unreal Engine with assets taken from its marketplace that is free for anyone to use. The game will have a proper 3D environment and the characters will be given proper 3D animation and behavior.

The proper working has not been decided yet but the A.I will have smart behavior, as it will be able to move on its own and even can patrol an area to find enemies to interact with. The reason for the game having its emphasis on the improvement of A.I is that many games that are released when unveiling the game attract the players or the customer with their graphics and world design but contain a lackluster or repetitive A.I. The player is in the start attracted to the game but quickly loses interest in the game as the game starts to feel too repetitive and boring. This leaves room for improvement which our game will try to improve upon.

Chapter 3

Software Specifications

3.1 Proposed System/Game

The project is created using the Unreal Engine 4, which is a game engine used for creating great open-world games with intelligent A.I. The game created will be using free assets that are available on the marketplace and will implement to achieve great results.

Like other games present ours will also have a proper storyline and graphics with proper animation that will make it stand out.

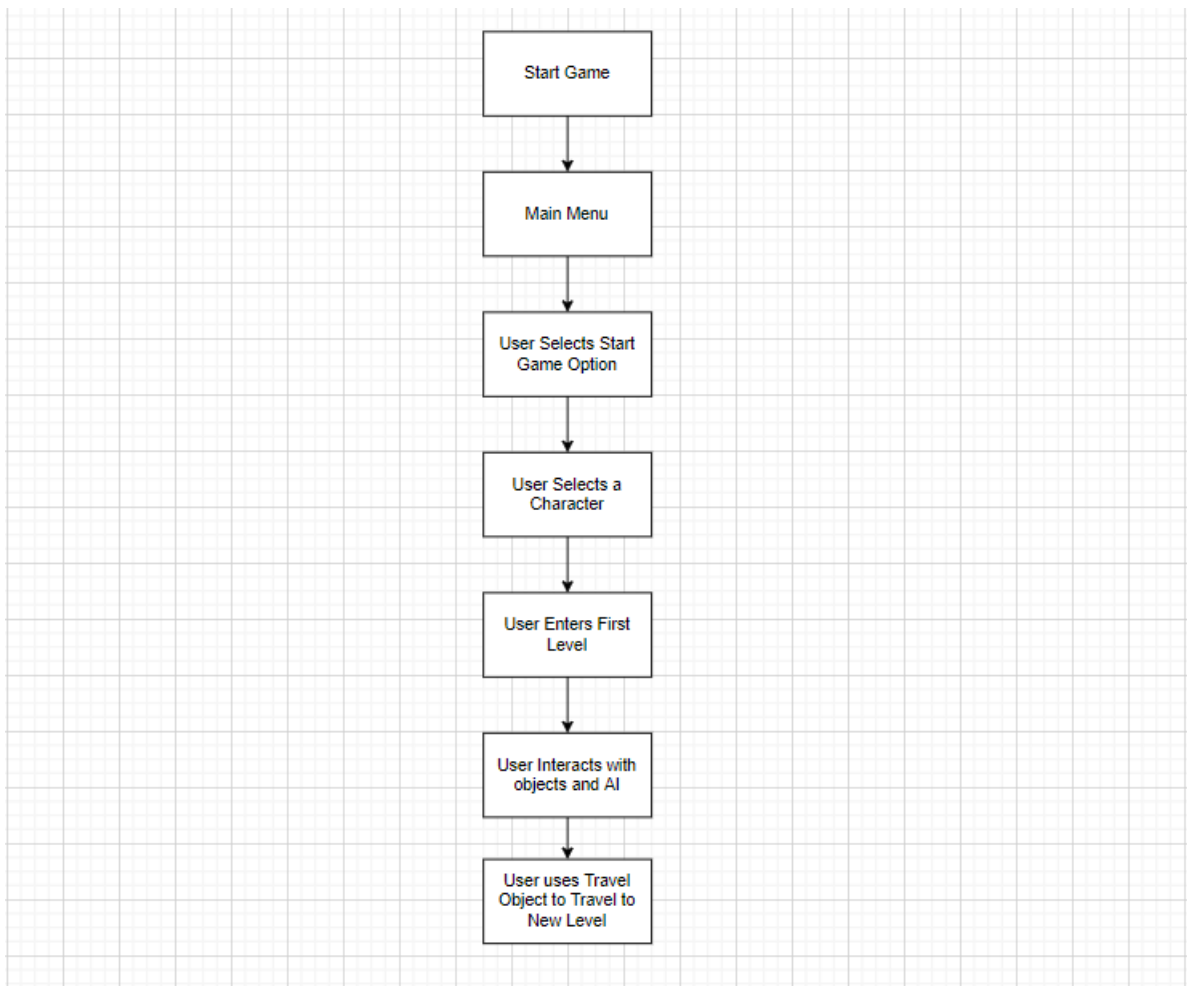


Figure 3.1: Flow of proposed system

3.2 Functional Requirement

3.2.1 Functional Requirement

Following are the functional requirement of the game:

1. New player must start the game using the New Game option in the main menu. The player must select a character who they wish to play as and will proceed with the game in a normal manner.
2. New player must start the game using the New Game option in the main menu. The player must select a character who they wish to play as and will proceed with the game in a normal manner.
3. The player can move the character with the 'WASD' keys, W for forward, S for backward, A for left movement, and D for right movement.

4. The player can use abilities implemented into the game using the numeric keys from 1 to 9 on the keyboard.
5. The player can change the camera angle using the mouse by moving it up, down, left, right, and combination.
6. The player character will take damage when attacked by the A.I. If the health variable reaches 0, the animation of the character being killed will be played.
7. When the player defeats an enemy A.I in the game they will gain experience points which after reaching a certain threshold will increase their character's level by one. After a level up the player can strengthen their character and unlock new abilities within the game further enhancing their experience.
8. The player can use different abilities in their selection bar. Using these abilities each has its own functionality which will either launch a projectile damaging the A.I being targeted or can give the character the user is using a certain amount of health of back using the magic points in the process.
9. The player can use different abilities in their selection bar. Using these abilities each has its own functionality which will either launch a projectile damaging the A.I being targeted or can give the character the user is using a certain amount of health of back using the magic points in the process.
10. When the player destroys an A.I in the game, they will gain an in-game currency that can be spent in the game.

3.2.2 Non Functional Requirements

1. Reliability:
The game being developed is reliable in terms of performance.
2. Performance:
On high-end computers, the game will give 40+ frames per second.
3. Usability:
The game UI is easy to use as it contains simple and easy-to-use functions.
4. Cost-effective:
If the game is released to the marketplace, it will be free to download and install.

3.3 Use Cases

The following use cases will provide a deeper understanding of the game and how it will function with the user.

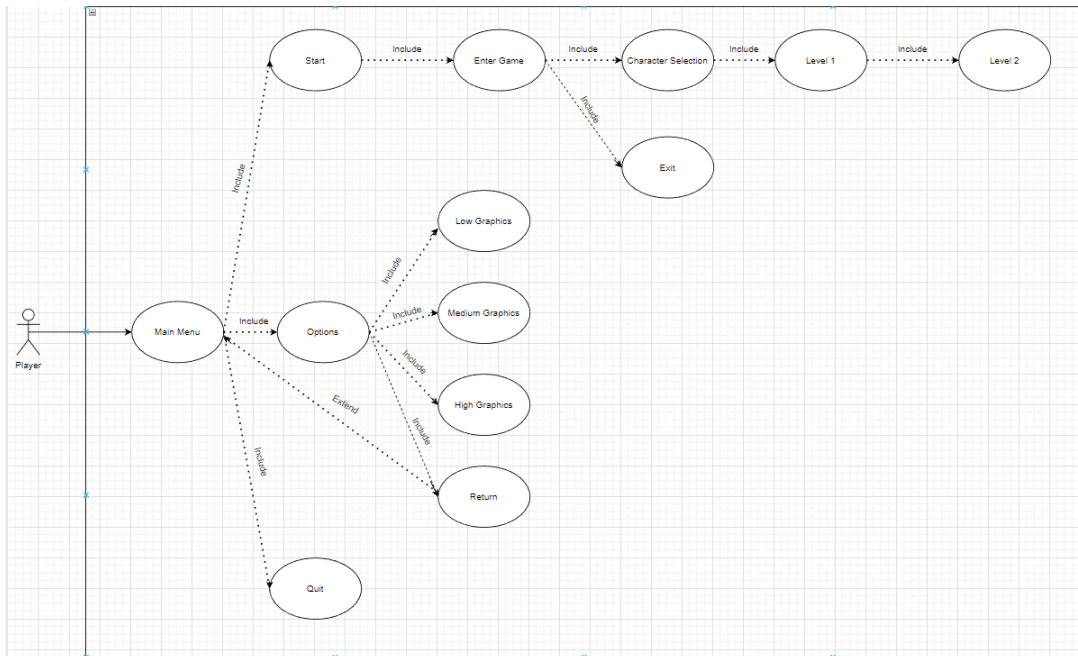


Figure 3.2: Use Case of Game Flow

Table 3.1: Flow of Game

Title	Game Flow
Use Case ID	AM01
Description	To start playing the game for the first time the user must follow the proper procedure.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must either be starting the game for the first time or wishes to start a new game.
Post Condition	Successful interaction with the game.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the application and choose an option from the main menu. 2. Player will choose the new game option when starting for the first time to choose and customize the character. 3. Player will choose the new game option when starting for the first time to choose and customize the character. 4. To continue where the player left off the the Load Game option will be used to select and load the game from the point where the user left off. 5. The options menu will let the player change the game settings such as sound volume or change the graphic settings.
Alternate Flow	<ol style="list-style-type: none"> 1. For wrong information if entered by the player, the player cannot change it as it is saved in the system and must create a new game file. 2. If the user is already playing the game, a message will pop up that the user already has a character and will now start a new game with a new character. 3. When starting for the first time if choosing the Load Game option, the slots will be empty.

Table 3.2: Main Menu Use Case

Title	Main Menu
Use Case ID	AM02
Description	How the player can interact with the main menu.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the Main Menu.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the application and choose an option from the main menu. 2. Player will choose the Start option to enter the game. 3. The options menu will let the player change the graphic settings ranging from low, medium, and high settings. 4. The player will choose the return option to return to the Main Menu. 5. The player will choose the Quit option to leave the game.
Alternate Flow	<ol style="list-style-type: none"> 1. The player was unable to open the game file due to an error.

Table 3.3: In-game Interface widget

Title	In-game Interface widget
Use Case ID	AM03
Description	How the In-game interface and widgets work
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. Player will choose a character and see if the correct character name is written. 3. The player will press the action keys to perform an action consuming mana and having the cooldown effect take place. 4. The player will interact with the hostile AI which will result in the health bar decreasing. 5. The player will use action abilities to destroy the hostile AI which will result in an increase in the experience bar. 6. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.4: Health Functionality

Title	Health Functionality
Use Case ID	AM04
Description	How the health functionality will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will take damage from the hostile AI resulting in the health bar decreasing 3. The player will use an ability to damage the hostile AI causing a hit which will decrease the health variable of the hostile AI and their destruction when reaching zero. 4. The player will return to the start when the health variable reaches zero. 5. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.5: Mana Functionality

Title	Mana Functionality
Use Case ID	AM05
Description	How the mana functionality will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will use an ability or spell which will decrease the mana variable. 3. The player will have to wait to use the ability again during the cooldown phase. 4. The player cannot use the abilities or spell when the mana variable is zero. 5. When the player uses the ability or spell when mana is less the required a pop-up will display that player has insufficient mana to use the ability. 6. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.6: Spell Casting Functionality

Title	Spell Casting Functionality
Use Case ID	AM06
Description	How the spell casting functionality will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will press the action keys to use a spell. 3. The player will lose a set number of the mana variable when using the abilities or spell. 4. The player cannot use the ability or spell when mana variable is below the required limit. 5. The hostile AI will receive damage to the health variable when hit with an ability or spell. 6. The player will recover a set amount of health variable when the heal spell or ability is used. 7. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.7: Health and mana regeneration.

Title	Health and mana regeneration.
Use Case ID	AM07
Description	How the Health and mana regeneration will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player when their health variable is below the maximum amount will regenerate a portion every second until the max amount is reached. 3. The player when their mana variable is below the maximum amount will regenerate a portion every second until the max amount is reached. 4. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.8: Health pickup

Title	Health pickup
Use Case ID	AM08
Description	How the Health pickup will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player when their health variable is below the maximum amount will regenerate a portion when walking over a health pickup. 3. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.9: Ability cooldown

Title	Ability cooldown.
Use Case ID	AM09
Description	How the Ability cooldown will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player uses an ability or spell which places the used ability or spell in a cooldown state. 3. The ability or spell icon will darken and become unavailable, entering a cooldown state. 4. When the cooldown period ends the player can use the ability or spell again. 5. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.10: Character movement and animation

Title	Character movement and animation
Use Case ID	AM10
Description	How the Character movement and animation will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player presses the movement key which moves the character. 3. The characters will play the proper animation when the movements are pressed. 4. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.11: Character selection

Title	Character selection
Use Case ID	AM11
Description	How the Character selection will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player can select a character from the character selection screen. 3. The character on the screen will be the one the player has chosen. 4. The character chosen by the player will have the proper animations which is allotted during the development phase. 5. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.12: Quest system

Title	Quest system
Use Case ID	AM12
Description	How the Quest system will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will receive a quest when walking over the quest object. 3. The player will see the current quest be updated after walking over the quest object. 4. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.13: Player leveling system

Title	Player levelling system
Use Case ID	AM13
Description	How the Player leveling system will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will receive experience after destroying a hostile AI. 3. The experience bar will increase after defeating a hostile AI. 4. The experience bar will return to zero after it has reached full capacity. 5. The character level will increase after the experience bar has reached full capacity and the integer type variable level shown at the top below the health and mana widget will be incremented by one. 6. The player will choose the close windows option to leave the game.
Alternate Flow	The player was unable to open the game file due to an error.

Table 3.14: Inventory system

Title	Inventory system
Use Case ID	AM14
Description	How the Inventory system will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will press the 'B' key which will open the inventory window. 3. The player will pick up an object which will transfer it to the inventory and will be seen by the player. 4. The player will not be able to pick up an object when inventory is full, and the object will remain on the ground or in the loot bag. 5. The player will choose the close windows option to leave the game.
Alternate Flow	<ol style="list-style-type: none"> 1. The player was unable to open the game file due to an error. 2. The inventory is already full and cannot store any more items.

Table 3.15: Loot bag system

Title	Loot bag system
Use Case ID	AM15
Description	How the loot bag system will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will destroy a hostile AI which will drop a loot bag. 3. The player will receive a set amount of the in-game currency gold coins and items when interacting with the loot bag. 4. The loot bag will disappear after the player interacts with it. 5. The gold coins and items picked up will appear in the inventory. 6. The player will choose the close windows option to leave the game.
Alternate Flow	<ol style="list-style-type: none"> 1. The player was unable to open the game file due to an error. 2. The player inventory was full and new items picked up did not show.

Table 3.16: Shop system

Title	Shop system
Use Case ID	AM16
Description	How the Shop system will work.
Primary Actor	Player.
Stake Holders	Player and company management
Pre-Condition	User must run the game using the executable file.
Post Condition	Successful interaction with the user widgets.
Basic Flow	<ol style="list-style-type: none"> 1. Player will enter the game and choose the Start option from the main menu. 2. The player will interact with the shop object which will open the shop interface. 3. The player can purchase items from the shop with gold coins. 4. The purchased items will appear in the character inventory. 5. The player will choose the close windows option to leave the game.
Alternate Flow	<ol style="list-style-type: none"> 1. The player was unable to open the game file due to an error. 2. The player does not have the required amount necessary to purchase the items.

Chapter 4

System Design

4.1 System Architecture

The proposed system of the project follows the 3-tier architecture as the game has its own logic and code that communicates with the presentation layer that shows the player graphical information that is understandable for them paired with the database to store the data. The 3-tier architecture is necessary for games as the game contains logic that is present on the user side.

4.1.1 Presentation Layer

The presentation layer is the layer the user will directly interact with and must be designed in such a way that it is easy for the player to understand and eye-catching as well to appeal to the player.

4.1.2 Logic Layer

The logic layer is the back end of the system where all the algorithms and functionality of the A.I and as well as the functions available for the player to use are available. These functions range from the movement of the character to the smart behavior of the A.I.

4.1.3 Database

This is the third layer of the 3-tier architecture in which the data of the games when saved is stored. This can be achieved by either attaching a database system or by using the given functionality of Unreal Engine itself, which are great choices.

4.1.4 Design Constraints

The restrictions that the player will have to follow for the system to function properly.

- The user is required to have a device with an operating system of Windows 7 or higher.
- User must have a graphical card or a high-end CPU with integrated graphics present in order to have a smooth experience.
- The game is only created to be run on Desktop or MAC computers.
- For input keyboard and mouse must be used as the game only accepts specific inputs.

4.2 Design Methodology

Currently, most of the development has shifted to agile methodology hence, therefore the project will greatly benefit from a tried and trusted approach.

The agile approach manages a project by dividing a huge project into smaller chunks of attainable goals. It includes a collaboration between stakeholders and developers to complete the project according to the requirements within the specified time. Every phase follows an incremental approach that follows a development cycle consisting of planning, executing, and evaluating.

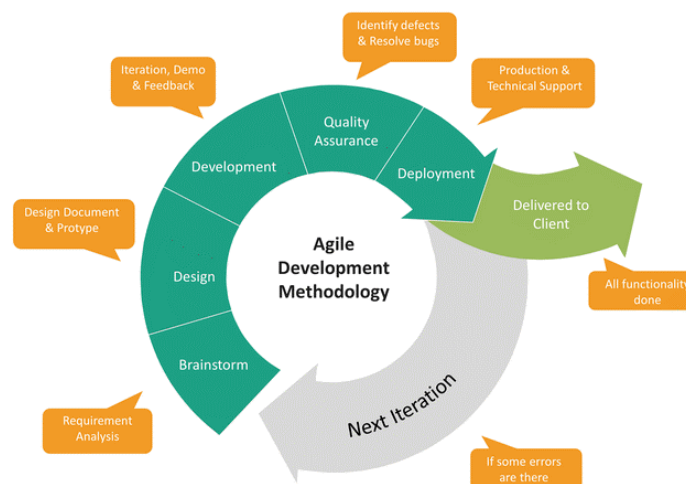


Figure 4.1: Agile Methodology Life Cycle

4.3 High Level Design

The high-level diagram contains the information about the system working. The main system body consists of three parts:

- Start Game
- AI Controller
- Character Events

The main part is the AI Controller which consists of the behavior tree which will define the behavior of the enemy character in the world.

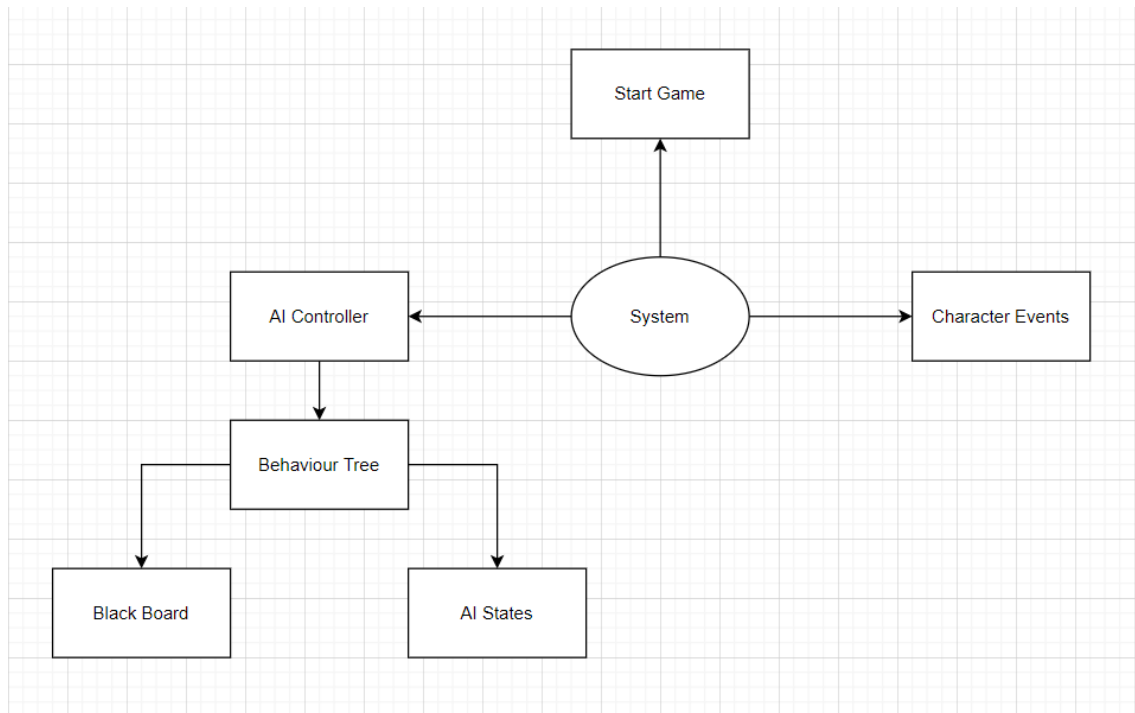


Figure 4.2: Block Diagram

4.4 Low-Level Design

The low-level diagram contains the structure or the flow of the game. The activity diagram contains the step-by-step process of how the user will interact with the game starting from the main menu to playing the game and saving of user characters level and experience points.

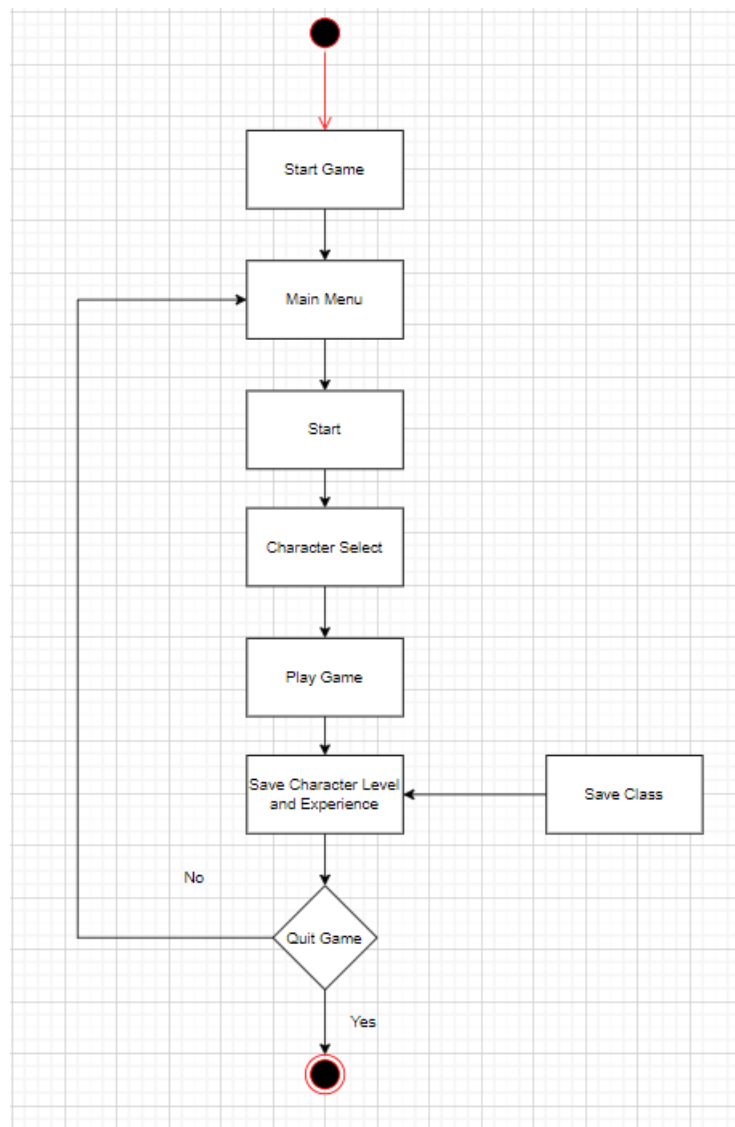


Figure 4.3: Activity Diagram

The sequence diagram explains the sequence of the game in how and what part of the game components are connected, such as the play game and the save class of the game.

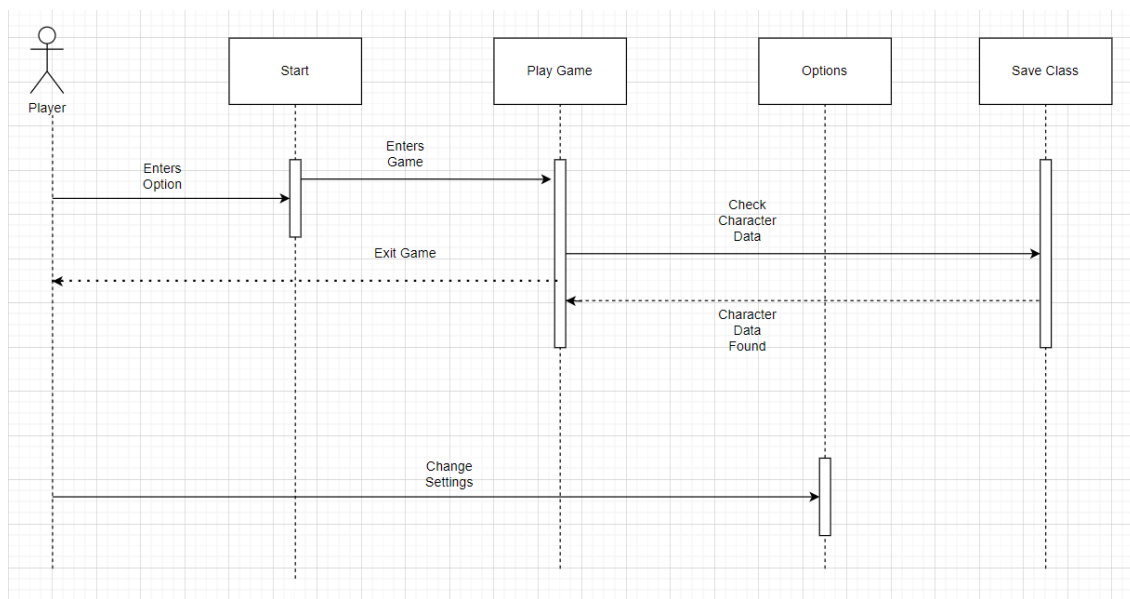


Figure 4.4: Sequence Diagram

4.5 A.I Implementation

The system or engine we are using for our game creation is the Unreal Engine 4. A game engine is a software that is primarily designed for the development of video games and generally, it includes relevant libraries and support programs. The term “engine” is similar to the term “software engine” used in the software industry. A game engine is a development software that utilizes a set of tools and features for game development. The core functionality typically provided by a game engine may include a rendering engine for 2D or 3D graphics, a physics engine, sound, scripting, animation, artificial intelligence, and so on.

The system or engine we are using for our game creation is the Unreal Engine 4. A game engine is a software that is primarily designed for the development of video games and generally, it includes relevant libraries and support programs. The term “engine” is similar to the term “software engine” used in the software industry. The game engine is a development software that utilizes a set of tools and features for game development. The core functionality typically provided by a game engine may include a rendering engine for 2D or 3D graphics, a physics engine, sound, scripting, animation, artificial intelligence, and so on.

Another great part about the reason why developers use the Unreal Engine is that you can either write the code from scratch or you can use blueprints given in the engine itself. Blueprints is a visual scripting system inside the engine and is a fast method to start prototyping the game, instead of having to write code line by line you can visually drag and drop nodes, set their properties in a UI, and drag wires to connect.

The implementation of the A.I in Unreal Engine 4 allows you to use a “body-soul-brain” pattern for the A.I.

- “Body” is the visual representation of a character or the actor/pawn that is physically present in the game.
- “Soul” is the entity that controls it at the moment.
- “Brain” what defines the behavior that the A.I will follow/behave in.
- “Memory” where it stores the information it needs to behave.

The A.I is made up of different components that work together, these include:

- A.I Perception component is used to registering different stimuli to the A.I ranging from sight and hearing which are a part of the sensing component.
- The environment query system is a feature that is used for collecting data about the environment.
- A behavior tree which is the most important part of giving life to the A.I. The behavior tree is a methodology that allows you to implement and view you’re A.I logic visually. It is a type of hierarchical task network resulting in a state-oriented design. The behavior tree is very extensive and is used to create an A.I with different behaviors such as the A.I will only attack the player character if its health variable falls below 50%, or if the health variable of A.I is about to reach 0% which will result in the A.I actor/pawn being destroyed the A.I will flee from the player to a safe distance.
- The navigation mesh is a tool that we can lay upon the terrain, which will tell the A.I the area it is allowed to move in. When we are setting the navigation mesh if any solid object enters the mesh the object if non-traversable will be a space where the A.I cannot enter and instead the A.I will move around it to reach its destination.

4.6 Algorithms used

Because the Unreal Engine contains blueprints and pre-defined functionality for the A.I such A.I perception and behavioral tree which is a graphical and mathematical model of a planned execution, not many algorithms are used in the engine. The only algorithm used is the following:

4.6.1 The A* Algorithm

Pathfinding is a system in which the A.I must find the shortest path between two points on the navigation mesh. The most commonly used algorithm was Dijkstra's Algorithm which was used to find the shortest path between two nodes, however, a variant of Dijkstra called the A* algorithm is used which does the same work however it can predict the cost of traversing nodes and is more efficient.

4.7 GUI Design

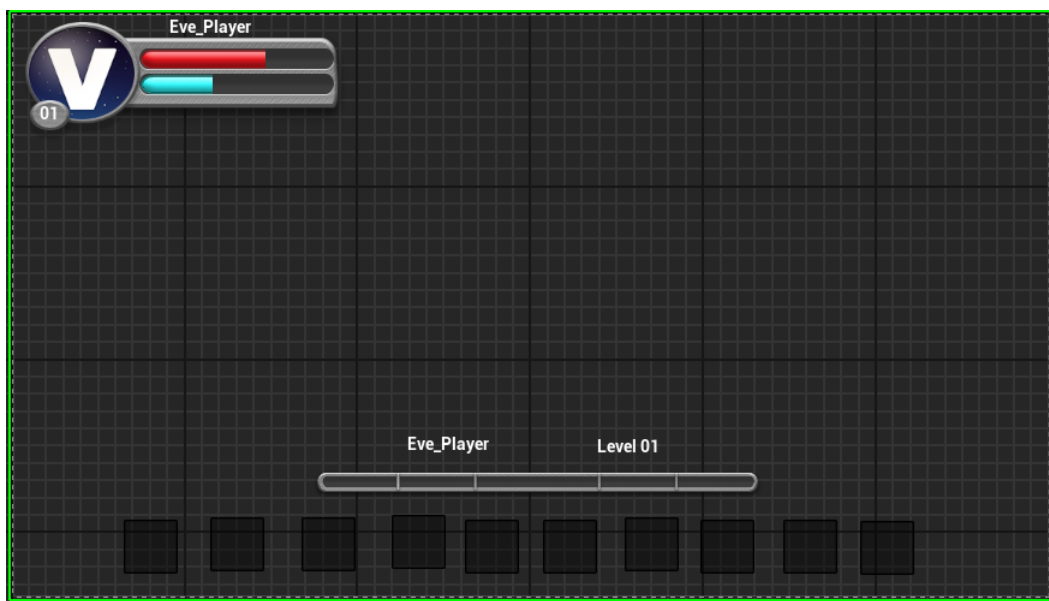


Figure 4.5: RPG HUD Display

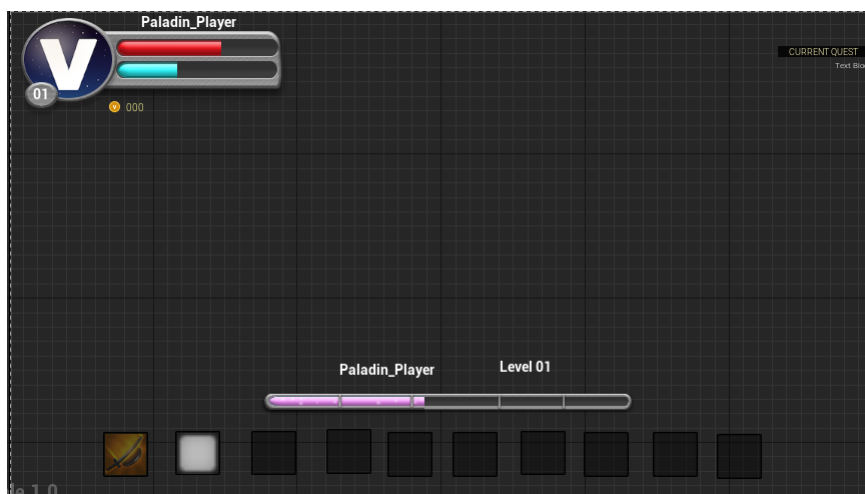


Figure 4.6: In Game View



Figure 4.7: In Game View with Roaming AI



Figure 4.8: Class Select Screen



Figure 4.9: Melee Character In Game View



Figure 4.10: In Game View with Inventory Accessed



Figure 4.11: In Game View with Item in Inventory after Item Pickup

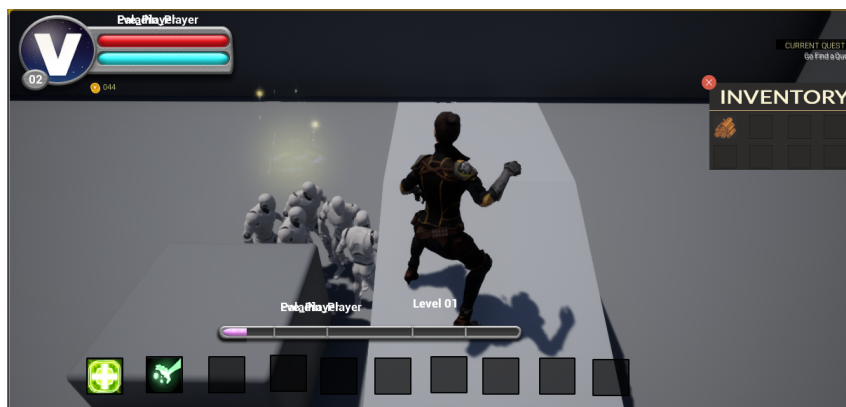


Figure 4.12: Loot Bag Screen

Chapter 5

System Implementation

The system implementation will define the technical specification and software components of how the game architecture works, what are its components, and what were tools, technologies, and methodology used to complete the project.

5.1 System Architecture

The system architecture will explain the internal and external components of the game. The proposed system includes:

5.1.1 The Video Game

The game will have only one module and that is the player/user. An admin is not needed as all changes are made separately in the game development software and applied to the game in a sort of patch or hotfix.

5.2 System Internal Component

The player will be the main actor who will be acting in the system as the game is an offline system software and an internet connection is not needed, therefore only one actor will exist.

5.2.1 Health

The health functionality is a system that will define the health variable of either the player character or the enemy A.I. It contains a health variable that will be different for the player and the A.I as well. It defines how much damage can be taken and when reaching the value of zero for the enemy A.I will destroy them and the player will show a game over the screen from which the user can either load from a checkpoint or quit the game.

5.2.2 Mana

The mana functionality is a system that is linked with the spell casting functionality. The player character is assigned a mana variable, and after each spell use a certain amount of mana amount will be used when reaching zero the player will not be able to use spell casting functionality and a no mana pop up will appear indicating that mana is zero until the mana regenerates or mana replenish item is picked up.

5.2.3 Spell Casting

The spell casting functionality is a system that is only available for the user to use. The spell casting uses a wide range of long-range magic attacks, player healing spells etc. This functionality is linked with the mana variable for activation. Each spell ability uses a different amount of the mana variable each time it is used.

5.2.4 Regeneration

The regeneration functionality is game difficulty balancing functionality that is linked with the health and mana variables. Through the use of health regeneration and mana regeneration variables, the health and mana variable will slowly fill back up in small amounts. This functionality exists to balance the game as the player will take damage from the enemy A.I when engaging in combat.

5.2.5 Item Pickup

The game will consist of two kinds of pick up that will increase the players health and mana variables respectively when standing close to them. The pick up item will increase the health or mana variable for a certain amount. This functionality is a game balancing functionality to make the game easier for the user to play.

5.2.6 Ability cooldown

The game will consist of two kinds of pick up that will increase the players health and mana variables respectively when standing close to them. The pick up item will increase the health or mana variable for a certain amount. This functionality is a game balancing functionality to make the game easier for the user to play.

5.2.7 Loot/Bag pickup item

The loot drops and pickup functionality is designed to reward for destroying/defeating the various enemy A.I across the map. When after defeating the enemy there will either always or a chance to drop an item that the player can use.

5.2.8 Melee combat system

The melee combat functionality is a system that the player will mainly use for fighting the enemy A.I at close range. By using weapons such as the sword the player can fight in close combat, the reason being that the spell casting abilities cannot be used repeatedly and making the melee combat the primary fighting method.

5.2.9 Character movements and animations

The character movement functionality allows the player to move across the map with an added bonus of proper animated movement for walking sideways and back. It also includes a sprinting system that will allow the user to run faster for faster terrain movement.

5.2.10 Character selection

The character selection functionality allows the player to choose different characters for their gameplay.

5.2.11 Quest System

The quest system functionality will allow the player to get quests that will reward the player upon completion. These quests can range from simple kill a certain amount of enemies to collect an item. At the end of each quest the player will be rewarded.

5.2.12 Player levelling

The player levelling functionality is created with the purpose of powering the player character as time goes on. When the player defeats an enemy A.I they will be rewarded

with experience points. This will allow the player to get stronger and defeat the enemy with ease.

5.2.13 Inventory

The inventory functionality is created with the purpose of giving the user to store the items the obtain from defeating the enemy. It acts like a bag that can keep items and is accessible by the user any time.

5.2.14 Shops

The inventory functionality is created with the purpose of giving the user to store the items the obtain from defeating the enemy. It acts like a bag that can keep items and is accessible by the user any time.

5.3 Tools and Technology

5.3.1 Unreal Engine

The inventory functionality is created with the purpose of giving the user to store the items the obtain from defeating the enemy. It acts like a bag that can keep items and is accessible by the user any time.

5.3.2 Microsoft Direct X

Microsoft direct X is a collection of application programming interfaces for handling tasks related to multimedia especially game programming. It contains a number of runtime libraries from the legacy direct X SDK. Almost all games created now a days require the direct X in order to function.

5.3.3 C++

C++ is a general-purpose programming language as an extension of C programming language.

5.4 Methodology

This game is developed using the agile methodology, because there are many functionalities that must be developed and incremented. There is only one actor to the game (player) so there not many requirements for many actors.

- **Phase 1** In the first phase we gathered the requirements that the game should have and implemented the basic functionalities.
- **Phase 2** In the second phase we developed a fully functional game with all the requirements and applied testing to ensure that no issues occur.

Chapter 6

System Testing and Evaluation

6.1 Introduction

System testing is the key to a successful system's working and performance. In order to ensure that all modules and functionalities work, and that no issues are caused at run time. These include:

- Graphical User Interface Testing
- Integration Testing
- Usability Testing
- Exception Testing
- Software Performance Testing

6.2 Graphical User Interface Testing

To ensure that the graphical UI works properly and has no issues.

Table 6.1: Main Menu UI Test Case

TC_No 1			
Main Menu			
Func_Req_No 1			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Open Game.	Pass or Fail	Pass
2	New Game option working.	Pass or Fail	Pass
3	Load Game option working.	Pass or Fail	Pass
4	About option working.	Pass or Fail	Pass
5	Settings option working.	Pass or Fail	Pass

Table 6.2: In-Game UI Test Case

TC_No 2			
In-Game user Interface			
Func_Req_No 2			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Action bar working.	Pass or Fail	Pass
2	Character Bar working.	Pass or Fail	Pass
3	Health and Manabar working.	Pass or Fail	Pass
4	Experience bar working.	Pass or Fail	Pass

6.2.1 Integration Testing

Table 6.3: Health Functionality Test Case

TC_No 2			
Health Functionality			
Func_Req_No 1			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Taking damage decreases the health.	Pass or Fail	Pass
2	Health reaching zero stops the game.	Pass or Fail	Pass
3	Enemy A.I takes damage to health.	Pass or Fail	Pass
4	Enemy A.I is destroyed when health reaches zero.	Pass or Fail	Pass

Table 6.4: Mana Function Test Case

TC_No 2			
Mana Functionality			
Func_Req_No 2			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Using spells decreases mana.	Pass or Fail	Pass
2	When mana is zero spells cannot be used.	Pass or Fail	Pass

Table 6.5: Spell Casting Test Case

TC_No 2			
Spell Casting			
Func_Req_No 3			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Pressing action keys uses a spell.	Pass or Fail	Pass
2	Using spells consumes mana.	Pass or Fail	Pass
3	When mana is zero or below the required amount it cannot be used.	Pass or Fail	Pass
4	Using a spell damages enemy A.I.	Pass or Fail	Pass
5	Using heal spell restores a certain amount of health variable.	Pass or Fail	Pass

Table 6.6: Health and Mana Regeneration Function Test Case

TC_No 2			
Health and Mana Regeneration			
Func_Req_No 4			
System Should have direct X			
Steps	Task	Expected Outcome	Result
1	Player health and mana are increased slowly overtime.	Pass or Fail	Pass
2	Regeneration amount can be increased after level up.	Pass or Fail	Pass

Table 6.7: Health and Mana Pickups Test Case

TC_No 2				
Health and Mana Pickups				
Func_Req_No 5				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	The health or mana pick up is visible.	Pass or Fail		Pass
2	The health or mana pick up can be in- teracted with.	Pass or Fail		Pass
3	The health or mana pick up increases its respective variable.	Pass or Fail		Pass
4	The pick up disappears after use.	Pass or Fail		Pass

Table 6.8: Ability Cooldown Test Case

TC_No 2				
Ability Cooldown				
Func_Req_No 6				
System Should have direct X				
Steps	Task	Expected Outcome	Result	
1	After using a spell a cooldown is placed.	Pass or Fail	Pass	
2	Spell cannot be used during cooldown.	Pass or Fail	Pass	
3	Cooldown period ends after a certain time.	Pass or Fail	Pass	
4	Spell can be used again after cooldown ends.	Pass or Fail	Pass	

Table 6.9: Character movement and animation Test Case

TC_No 2				
Character movement and animation				
Func_Req_No 7				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Pressing a movement key moves the player character.	Pass or Fail		Pass
2	Pressing a movement key moves the player character.	Pass or Fail		Pass
3	Pressing the sprint action key makes the player character faster.	Pass or Fail		Pass
4	The pick up disappears after use.	Pass or Fail		Pass

6.3 Exception Testing

To ensure that the system will not crash if any run time error occur.

Table 6.10: In-Game Functionality Exception Test Case

TC_No 3				
In-Game Functionality				
Func_Req_No 2				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Use a spell.	Pass or Fail		Pass
2	Use a Spell immediately after using and see if it works.	Pass or Fail		Pass
3	Use a spell when the mana variable is zero and see if it works.	Pass or Fail		Pass

Table 6.11: Character Movement and Physics Test Case

TC_No 2				
Character Movement and Physics				
Func_Req_No 3				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Move the character.	Pass or Fail		Pass
2	Try to move through a solid object and see if works.	Pass or Fail		Pass
3	See if a solid object can be moved through player movement.	Pass or Fail		Pass

6.4 Performance Testing

To ensure that the system is efficient and will respond to how the user wants.

Table 6.12: Overall Game Performance Test Case

TC_No 2				
Main Menu				
Func_Req_No 3				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Open the game.	Pass or Fail		Pass
2	Start a new game.	Pass or Fail		Pass
3	Explore the world through movement.	Pass or Fail		Pass
4	Defeat an enemy A.I.	Pass or Fail		Pass
6	Use a Spell.	Pass or Fail		Pass
10	Exit the game.	Pass or Fail		Pass

Table 6.13: Character Selection Test Case

TC_No 2				
Character Selection				
Func_Req_No 10				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	The player can select a different character when starting a new game.	Pass or Fail		Pass
2	The character chosen is one the player can played and another character.	Pass or Fail		Pass
3	The chosen character has proper animations and movements.	Pass or Fail		Pass

Table 6.14: Quest System Test Case

TC_No 2				
Quest System				
Func_Req_No 11				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	The quest system can be accessed.	Pass or Fail		Pass
2	The quest can be completed.	Pass or Fail		Fail
3	The quest gives a reward upon completion.	Pass or Fail		Fail

Table 6.15: Player Levelling Test Case

TC_No 2				
Player levelling				
Func_Req_No '12				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	After the experience variable reaches a certain amount the player can level up.	Pass or Fail		Pass
2	Experience can be obtained after destroying an enemy A.I.	Pass or Fail		Pass
3	The experience bar will return to zero after a level up.	Pass or Fail		Pass

Table 6.16: Player Inventory Test Case

TC_No 2				
Inventory				
Func_Req_No 13				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	The player inventory can be accessed.	Pass or Fail		Pass
2	Items are placed in the inventory after pickup.	Pass or Fail		Pass
3	Items can be removed from player inventory.	Pass or Fail		Fail

Table 6.17: Shop Test Case

TC_No 2				
Shop				
Func_Req_No 14				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	The shop menu can be accessed.	Pass or Fail		Pass
2	Items can be purchased using in-game currency.	Pass or Fail		Pass

6.5 Exception Testing

To ensure that the system will not crash if any run time error occur.

Table 6.18: Main Menu Exception Test Case

TC_No 3				
Main Menu				
Func_Req_No 1				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Press Load game option.	Pass or Fail		Fail
2	Try to load a game when no save file is present and see if it works.	Pass or Fail		Fail

Table 6.19: In-Game Spell Exception Test Case

TC_No 3				
In-Game Functionality				
Func_Req_No 2				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Use a Spell..	Pass or Fail		Pass
2	Use a Spell immediately after using and see if it works.	Pass or Fail		Pass
3	Use a spell when the mana variable is zero and see if it works.	Pass or Fail		Pass

Table 6.20: Character Movement and Physics Test Case

TC_No 3				
In-Game Functionality				
Func_Req_No 3				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Move the character.	Pass or Fail		Pass
2	Try to move through a solid object and see if works.	Pass or Fail		Pass
3	See if a solid object can be moved through player movement.	Pass or Fail		Pass

6.6 Performance Testing

To ensure that the system is efficient and will respond to how the user wants.

Table 6.21: Overall Game Performance Test Case

TC_No 4				
Main Menu				
Func_Req_No 1				
System Should have direct X				
Steps	Task	Expected come	Out-	Result
1	Open the game.	Pass or Fail		Pass
2	Start a new game.	Pass or Fail		Fail
3	Explore the world through movement.	Pass or Fail		Pass
4	Defeat an enemy A.I.	Pass or Fail		Pass
5	Pick up an item.	Pass or Fail		Pass
6	Use a Spell.	Pass or Fail		Pass
7	Complete a quest.	Pass or Fail		Fail
8	Access inventory.	Pass or Fail		Pass
9	Access shop and make a purchase.	Pass or Fail		Pass
10	Exit the game.	Pass or Fail		Pass

Chapter 7

Conclusions

7.1 Conclusions

The end result of the project is a game that can be run on the computer system. This project is an entertainment-based project targeted at all audiences. The game contains easy-to-understand functions that are used in modern games that most users are familiar with. With the addition of enemy A.I that interacts with the user when seen. All the processes present in the game have been added and implemented manually from start to finish. Developing this project has been a long journey with its own rewards. During the time spent developing this project, the tried and tested agile methodology has been used and has used modern skills and tools that are in the practice in many industries.

7.2 Future Enhancements

- **Crafting System**

The game can be enhanced by adding a crafting system. By defeating the enemy A.I they will drop items that can be used to create different items such as weapons and so on.

- **Potion making**

The game will also include a potion crafting system that will allow the player to brew and create different potions that can either help the player.

Appendix A

User Manual

A.1 Introduction

The game is a third-person open-world game that gives the user the freedom to move on a large map. The game also contains enemy characters that will damage the player resulting in the user character's death if the health bar reaches zero after which will be sent to the start of the first level. The game also contains a character selection screen that will let the user choose between two different characters.

A.2 System Requirements

- Desktop PC or Mac.
- Windows 7 64-bit or Mac OS X 10.9. 2 or later.
- Quad-core Intel or AMD processor, 2.5 GHz or faster.
- NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series card or higher.
- 8 GB RAM.

A.3 Installation

The game is available in an executable file. The user only has to copy the game folder and can run the game from the '.exe' file.

A.4 Settings

The game contains different settings for the graphical interface. The game contains three options:

- Low Graphics
- Medium Graphics
- High Graphics

Choosing any of the above options will change the graphical settings of the game.

A.5 Controls

For the movement of the user character the controls are as follows:

- Pressing the 'W' key will move the user character in the forward direction.
- Pressing the 'S' key will move the user character in the backward direction.
- Pressing the 'A' key will move the user character in the left direction.
- Pressing the 'D' key will move the user character in the right direction.
- Pressing the '1' key will use the first ability in the action bar.
- Pressing the '2' key will use the second ability in the action bar.
- Pressing the 'Space' key will allow the user character to jump.
- Moving the mouse will move the camera of the user character.

Bibliography

- [1] "J. r. r. tolkien,," [Online]. Available: https://en.wikipedia.org/wiki/J._R._R._Tolkien
Cited on pp. 1 and 3.
- [2] "Project management guide." [Online]. Available: <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>.
Cited on p. 3.
- [3] X. Y. B. Xia, "Recent research on a.i.," p. 6, 2020." Cited on p. 3.
- [4] S. L.-U. H. M.-A. B. D.-A. P. G.-C. A. Sanchez-Ruiz, "Game ai for a turn-based strategy game with plan adaptation and ontology-based retrieval," p. 8." Cited on p. 4.
- [5] M. Thielmeyer, "'forbes,'" [Online]. Available: <https://www.forbes.com/sites/maxthielmeyer/2019/03/07/rocket-leagues-smartest-artificial-intelligence-bots-are-facing-off-this-weekend/?sh=9ed9c546e5d5>. Cited on p. 7.
- [6] D. Tangness, "davidtangness.com." [Online]. Available: <http://www.davidtangness.com/journal/2014/11/6/game-analysis-guard-ai-in-splinter-cell-blacklist>. Cited on p. 8.
- [7] S. Horti, "Rockpapershotgun,," [Online]. Available: <https://www.rockpapershotgun.com/why-fears-ai-is-still-the-best-in-first-person-shooters>. Cited on p. 8.
- [8] AIandGames, "'the ai of alien: Isolation | ai and games," 2016." Cited on p. 9.