



MAHPARA ZAMID
01-235192-036
SANAWAR IQBAL
01-135201-091

Wandering Spirit Game

Bachelor of Science in Information Technology

Supervisor: Dr.Muhammad Asfand-e-yar

Department of Computer Science
Bahria University, Islamabad

October 2023

Certificate

We accept the work contained in the report titled 'Wandering Spirit Game,' written by Mrs. Mahpara Zamid and Mr. Sanawar Iqbal, as confirmation of the required standard for partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor: Dr.Muhammad Asfand-e-yar

Internal Examiner: Name of the Internal Examiner (Title)

External Examiner: Name of the External Examiner (Title)

Project Coordinator: Dr.Faisal Imran

Head of the Department: Dr Arif ur Rehman

October 2023

Abstract

"The Wandering Spirit" is a story-based 3D strategy game where the main character is a baby ghost searching for his parents in a human neighborhood. The game offers various survival options and techniques within an interactive graphical environment. It is designed to be accessible to players of all levels of experience. The game consists of four different levels, each with unique tasks and objectives to complete. Players can freely explore the game environment from a third-person perspective, hide the character from humans for survival, feed the ghost character to progress to the next level, and enable the character to vanish from sight. Game functionality includes character movement controlled via keyboard and mouse commands, the ghost's ability to consume food, and the capability to disappear from the view of humans. The game immerses the user in a realistic third-person perspective, enhancing their survival, problem-solving, and strategic skills.

Acknowledgments

In the Name of Allah, the Most Merciful, the Most Beneficent. Praise be to the Lord of all worlds. Peace and blessings be upon our Prophet, Muhammad, his family, and all of his companions. I would like to acknowledge and express my warmest thanks to my supervisor, Dr. Muhammad Asfand-e-yar, who made this work possible. His guidance and advice carried me through all the stages of writing my project. I would also like to thank my university faculty for making my defense an enjoyable moment, and for your brilliant comments and suggestions. Thanks to all of you.

MAHPARA ZAMID, SANAWAR IQBAL
Islamabad, Pakistan

October 2023

"You learn more from failure than from success. Don't let it stop you. Failure builds character"

Contents

Abstract	i
1 Introduction	1
1.1 Introduction	1
1.2 Background	1
1.3 Objectives	2
1.4 Project Description	2
1.5 Methodology	2
1.5.1 The Iterative Model	3
1.6 Features	3
1.7 Project Scope	3
1.7.1 Introduction	3
1.7.2 Goals	4
1.7.3 Technologies	4
1.8 Limitations	4
1.9 Feasibility Study	4
1.9.1 Risks Involved	4
1.9.2 Resource Requirements	4
1.9.3 Programming Language	5
1.9.4 3D Characters	5
1.10 Introduction to Unity Engine	5
2 Literature Review	6
2.1 Existing Games	6
2.1.1 Survival Game	6
2.1.2 Rust	6
2.1.3 Resident Evils	7
2.1.4 Subnautica	7
2.1.5 Green Hill	8
3 Requirement Specifications	9
3.1 Purpose	9
3.2 System Overview	9
3.3 Requirement Gathering and Analysis	9
3.4 Design	9
3.5 Implementation	9
3.6 Testing	10
3.7 Working	10
3.8 Process Modeling	10
3.9 Functional requirements	10
3.9.1 Start	10
3.9.2 Load	11

3.9.3	Setting	11
3.9.4	Exit	11
3.10	Non-Functional Requirements	11
3.10.1	Cross-Platform Compatibility:	11
3.10.2	Regulatory Compliance:	11
3.10.3	Load Balancing:	12
3.10.4	Security Updates and Patching:	12
3.10.5	Environmental Simulation:	12
3.11	Use case modeling and Description	12
3.12	System use case diagram for Wandering Spirit	13
3.13	Use case description for all scenarios Wandering Spirit	14
3.13.1	Use case description for Start game	14
3.13.2	Use case description for Load game	14
3.13.3	Use case description for setting	15
3.13.4	Use case description for quit game	15
3.13.5	Start Game Use-Case Description	16
3.13.6	Load Game Use-Case Description	16
3.13.7	Setting Game Use-Case Description	17
3.13.8	Exit Game Use-Case Description	17
4	Design	18
4.1	System Architecture	18
4.1.1	Logical Viewpoint	18
4.1.2	Explanation	20
4.2	Sequence diagram	20
4.3	Composite View	22
4.4	Information Viewpoint	23
4.5	State Dynamic view	24
4.6	Algorithmic Viewpoint	26
5	System Implementation	27
5.1	Software Requirements	27
5.2	System Components	27
5.3	Front end	27
5.4	Game Environment in unity	28
5.5	Other Options	30
6	System Testing and Evaluation	38
6.1	Chapter Overview	38
6.2	Testing and Methodologies and Techniques	38
6.3	Test Cases	39
6.3.1	Character Movement Test Case	39
6.3.2	3rd Person User Perspective Test Case	39
6.3.3	Eating Ability Test Case	40
6.3.4	Vanish ability	40
6.3.5	Select Option from Menu Test Case	40
6.3.6	New Game Test Case	41
6.3.7	Pause Option	41
6.3.8	Exit Option Test Case	42
6.3.9	Hurdles Test Case	42
6.3.10	Game should have hurdles for players	42
6.3.11	Player must complete the level	43

6.3.12	Detecting ghost near Old-man character	43
6.3.13	Detecting ghost near Old-man character	43
6.3.14	Player should be able to change the volume level of game	44
6.3.15	Game should be in 3D environment	44
6.4	User Manual	44
6.4.1	Introduction	45
6.4.2	Hardware and software requirement for system	45
6.4.3	Installation Guide for Application	45
6.4.4	Operational Manual	45
7	Conclusions	52
7.1	Future Work	52

List of Figures

2.1	Rust game	6
2.2	Evil dead	7
2.3	Subnautic	8
2.4	Green Hill	8
3.1	System use-case diagram	13
3.2	Start-game use-case diagram	14
3.3	Load-game use-case diagram	14
3.4	setting use-case diagram	15
3.5	Quit-game use-case diagram	15
4.1	Logical Diagram	19
4.2	Sequence Diagram	21
4.3	Composite View Diagram	22
4.4	Information Viewpoint	23
4.5	State dynamic view	25
5.1	Game Environment	28
5.2	Game Environment	29
5.3	Character Interface	30
5.4	Character Side Interface + Animation	31
5.5	Man Character Model	31
5.6	Animal Character Model	32
5.7	House Asset Front	32
5.8	House Asset Side Interface	33
5.9	game Scripts	33
5.10	Man Character Animations	34
5.11	dog in environment	34
5.12	Introduction Video of Game	35
5.13	1st Level Interface	35
5.14	Eating food	36
5.15	Character Vanished in Trash-Bin From Old Man	36
5.16	Character Running	37
6.1	Game main menu	46
6.2	Main Menu New Game Button	47
6.3	Main Menu Load Screen Button	48
6.4	Main Menu Setting Button	48
6.5	Setting Screen	49
6.6	Main Menu Exit Button	49
6.7	Ending interface	50
6.8	Story button interface	50
6.9	user controller guide	51

List of Tables

3.1	Use Case: Start Game Use-Case Description	16
3.2	Use Case: Load Game Use-Case Description	16
3.3	Use Case: Setting Game Use-Case Description	17
3.4	Use Case: Exit Use-case Description	17
6.1	Test Case TC-1: Character Movement Test Case	39
6.2	Test Case TC-2: 3rd Person User Perspective Test Case	39
6.3	Test Case TC-3: Eating Ability of Ghost Test Case	40
6.4	Test Case TC-4: Vanish Ability Test-case	40
6.5	Test Case TC-5: Select Option from Menu TestCase	41
6.6	Test Case TC-7: Pause Option	41
6.7	Test Case TC-8: Exit Option TestCase	42
6.8	Test Case TC-9: Hurdles Test Case	42
6.9	Test Case TC-10: Hurdles Test case(2)	42
6.10	Test Case TC-11: Completing a Level (Failure Condition)	43
6.11	Test Case TC-12: Old Man Character Test Case	43
6.12	Test Case TC-13: Old Man Character test Case(2)	43
6.13	Test Case TC-14: change volume of game	44
6.14	Test Case TC-15: Game should be in 3D environment	44

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
MVC	Model-View-Controller
OS	Operating System
SE	Software Engineering
SQL	Structured Query Language
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Introduction

This chapter serves as the introduction to the listing of some key game specifications for our final year project and outlines certain target game mechanics. By reviewing a few models that describe these game mechanics, you will gain a good understanding of how they work. Lastly, we will review the character control scheme for the game. While endless runner games often provide a sense of relaxation and freedom to the user, creating such a game entails numerous challenges and the need for a substantial amount of game content. Our project is a game named 'Wandering Spirit.' 'Wandering Spirit' is a comprehensive strategy game featuring multiple levels. The game's main character is a young ghost who loses its parents in a human neighborhood. This baby ghost is afraid of people and hasn't yet mastered many ghostly tricks, making it difficult for it to reunite with its parents. Now, it must find food and remain concealed in a bustling town. The game's primary mission is to employ logic to rescue the ghost.

1.2 Background

"Wandering Spirit" is a story-based 3D strategy game where the main character is a ghost on a quest to find its parents in a human neighborhood. The game offers a free-to-play environment with a wide range of survival options, strategies, and interactive graphics. The game consists of four distinct levels, each featuring unique tasks and objectives. Users can freely explore the game environment from a third-person perspective, conceal the character from people to ensure survival, feed the ghost character to progress to the next level, and enable the character to become invisible to humans. Game functionality includes character movement controlled through keyboard and mouse commands, a third-person perspective for the user, the ghost's ability to consume food, and the capability to become invisible to humans. The game's significance lies in its potential to enhance cognitive abilities and learning skills, improve emotional management, and refine strategic thinking. It is a standalone game with no prior versions and is compatible with Windows operating systems.

1.3 Objectives

Survival games are a popular genre of video games that typically involve the player navigating a hostile environment and trying to stay alive as long as possible. The primary objective of survival games is to test a player's ability to adapt, strategize, and make decisions in a challenging and often unforgiving setting. The specific objectives in survival games can vary but generally include the following:

- **Exploration:** Players often need to explore the game world to find vital resources, safe locations, and discover the story or objectives. Exploring new areas can be dangerous but is often necessary.
- **Crafting:** Crafting is a core element in many survival games. Players can create tools, weapons, and equipment from the resources they gather. Crafting allows for self-sufficiency and customization.
- **Base Building:** Some survival games involve building and fortifying a base or shelter. This serves as a haven against threats and a place to store resources.
- **Defeating Enemies:** Depending on the game, players may have to combat hostile creatures or other players. Defeating these threats can provide valuable resources and enhance safety.

Our game has the following objectives, as it primarily focuses on animation and storytelling. The major objectives are as follows:

- To help players overcome their fears and phobias.
- To create a gameplay mode that is accessible to everyone.
- To provide a high-quality environment.
- To assist individuals in honing their problem-solving and strategy-making abilities.

1.4 Project Description

As developers, we take pride in the inclusion of the baby ghost character in our game, as it adds a crucial dimension to the story line and enhances the overall gaming experience[1]. Players will face the challenge of navigating a human neighborhood while evading the haunting presence of the baby ghost. We are dedicated to continuously improving and refining this game to ensure it delivers the most engaging and educational experience possible. In the future, we plan to launch our game on the Google Play Store, expanding our reach to a broader audience and generating revenue to support ongoing development. We firmly believe that our game offers a unique and valuable opportunity for players to develop critical thinking skills, emotional intelligence, and problem-solving abilities, all while enjoying themselves and embarking on a thrilling adventure. So, prepare to confront the challenges posed by the baby ghost in the game and put your skills to the test for survival in the human neighborhood[2].

1.5 Methodology

The methodology that we will be using for our project is the iterative model.

1.5.1 The Iterative Model

The iterative model is a software development methodology in which the software development life cycle is divided into smaller iterative cycles or stages. Each iteration involves a series of steps, including planning, requirements gathering, design, implementation, testing, and evaluation. In the iterative model, the development process starts with a minimal set of requirements and functionalities, which are gradually expanded in each iteration. Each iteration produces a working software product that can be tested and evaluated. Based on the results of the evaluation, the development team can then refine the software's requirements, design, and implementation in the next iteration. The iterative model is a flexible approach to software development that allows the development team to adjust the project plan based on feedback and changes in requirements. It also enables a faster development cycle and can result in a higher-quality product due to continuous testing and evaluation throughout the development process.

1.6 Features

Main features of the Wander Spirit game are:

- Single player.
- 3D graphic.
- Baby ghost character movement.
- Level up after meeting certain conditions.
- Users can start new game.
- User Stages will be stored in the database.
- Food eating by baby ghost to enhance his energy and level up.
- Baby ghosts vanish ability.
- 3D Background scenery
- Power ups such as eating food, coins etc.
- Character will vanish is he have interaction with human.

1.7 Project Scope

1.7.1 Introduction

This software will be an entertainment game intended for leisure purposes. It is designed as a single-player game with defined gameplay boundaries, allowing only one person to play the game at a time.

1.7.2 Goals

Through this project, we aim to achieve the following objectives:

- Develop practical skills such as problem-solving, critical thinking, and project management, which are essential in many industries.
- Enhance communication skills through the presentation of project findings and results to peers, faculty, and industry professionals.
- Showcase creativity and innovation by creating a unique and original project idea.
- Improve our ability to work effectively in a team.

1.7.3 Technologies

The tools and Technologies we will be using are:

- Unity Engine
- Visual Studio
- Maya
- Adobe Photoshop

1.8 Limitations

A few limitations of our project are:

Limited Scope: Endless runner games typically have limited scope and gameplay mechanics, potentially limiting their appeal to certain players.

1.9 Feasibility Study

1.9.1 Risks Involved

There is nothing in the world that is risk free, with there:

- Risk of our game crashing
- Improper implementation of our idea
- Can be done in available time?

1.9.2 Resource Requirements

- Computer or Laptop
- Window and other operating systems
- Ram 4 GB or above

1.9.3 Programming Language

All the programming of our game will be done using C programming language. C is a general-purpose, multi-paradigm programming language. C encompasses static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.

1.9.4 3D Characters

For 3D character design we will use blender software to design 3D characters like ghosts.

1.10 Introduction to Unity Engine

Unity is a cross-platform game engine initially released by Unity Technologies in 2005. Unity's primary focus lies in the development of both 2D and 3D games as well as interactive content[3]. Currently, Unity supports over 20 different target platforms for deployment, with its most popular platforms being PC, Android, and iOS systems. In our project, Unity will be used in several ways, including:

- **Game Design:** Unity's built-in game design tools, such as the visual editor, can be utilized to create a game world, design levels, and customize the game's user interface.
- **Asset Importing:** Unity supports various file formats for importing 2D and 3D assets, including sprites, models, textures, and animations. These assets can be employed to create characters, obstacles, power-ups, and other game elements.
- **Physics Engine:** Unity's physics engine can simulate realistic movements and interactions between game objects, including gravity and collisions.
- **Testing:** Unity provides various testing tools, such as the Unity Test Runner and Profiler, which can be employed to test game mechanics, optimize performance, and ensure the game runs smoothly on different devices.

Chapter 2

Literature Review

2.1 Existing Games

2.1.1 Survival Game

While there are numerous survival games available on the internet, such as Rust and Resident Evil, our game focuses solely on survival through shooting.

2.1.2 Rust

"Rust" is a popular multiplayer survival video game developed and published by Face Punch Studios[4]. Initially launched in early access in 2013, it officially reached its full release in 2018. This game boasts compatibility with multiple platforms, including Windows, macOS, Linux, PlayStation, and Xbox. In "Rust," the primary goal is to survive in a challenging wilderness by gathering or, at times, pilfering essential materials for sustenance and protection. To thrive, players must effectively manage their hunger, thirst, and overall health, as neglecting these vital aspects can lead to a swift demise. Notably, while the game presents menacing creatures such as bears and wolves, the most significant danger stems from fellow players, given that "Rust" is exclusively a multiplayer experience. Overall, "Rust" delivers an intensely challenging and dynamic survival experience, placing a heavy emphasis on resource management, strategic decision-making, and the intricate interactions among players. However, please bear in mind that my knowledge is based on information available up until September 2021, and there may have been updates or changes to the game since that time[5].



Figure 2.1: Rust game

2.1.3 Resident Evils

The "Resident Evil" is a highly popular and influential survival horror video game series developed by Capcom[6]. The series originated in 1996 with the release of the first game for the PlayStation console. The games are known for their focus on horror, exploration, puzzle-solving, and survival mechanics. The main premise of the "Resident Evil" series revolves around various outbreaks of a zombie virus or other biohazards, often set in a fictional universe where secretive and unethical activities of powerful corporations lead to disastrous consequences. Players typically take on the role of characters who are tasked with investigating these outbreaks, navigating through dangerous environments, solving puzzles, and confronting various monsters and enemies.



Figure 2.2: Evil dead

2.1.4 Subnautica

"Subnautica" is an exciting game where the players find themselves stranded in an underwater world on an alien planet[7]. The user needs to explore the deep ocean, collect things they can use, and build places to stay and submarines to get around. But watch out for the different creatures in the water, some are friendly, and others are not! As the user explores, they'll uncover the secrets of this mysterious world. "Subnautica" is a game that mixes adventure, survival, and storytelling, all set in a beautiful but sometimes dangerous underwater world. It challenges the user to be clever and brave as they navigate this amazing alien ocean.



Figure 2.3: Subnautic

2.1.5 Green Hill

"Green Hell" is indeed an adventurous survival game set in the Amazon rain forest, where players must survive the challenging environment by building shelter, finding food, and maintaining their health while exploring the jungle[8]. The mention of wild animals and hidden dangers adds to the sense of danger and adventure in the game. You have also highlighted the progression aspect, where players uncover mysteries about the jungle and their presence there, which adds depth to the game play.



Figure 2.4: Green Hill

Chapter 3

Requirement Specifications

3.1 Purpose

The purpose of this chapter is to explain in detail about the design and architecture of the system specification. In this chapter the details about the system will be defined and the entire system design will be explained.

3.2 System Overview

The design regarding the system, functionality of the parts, communication between unity tools and technology, Integrated development environment (IDE)/Languages used are clarified and illustrated.

3.3 Requirement Gathering and Analysis

At this stage we will gather information about other games that are related to our game but are already present on the google play store, after this the information about the gaming engine will be gathered and the required specification is conducted to proceed further.

3.4 Design

In the design phase we would determine which modelling technique to follow and how to design the basic model of our game including the design of the characters, background scenery etc.

3.5 Implementation

In the implementation or development phase the actual coding of all the levels will be conducted based on design developed in the design phase of the project. Here programming will be done using visual studio IDE for Unity scripting.

3.6 Testing

At this stage our project will be tested by creating different test cases to see if our project is functioning properly or not.

3.7 Working

Making any kind of game is a challenge on its own. We were just determined to make this project so that we could come up with a good project that would help us improve our coding capabilities and would be a good opportunity for us to grow and learn. Some of the challenges that we faced are as following:

- Set up game environment with different textures and unity tools.
- Getting familiar with unity asset store for the creation and importing of our characters.
- Then for some basic editing of the textures and creating some 3D Character will used. We learn basic Blender for this purpose.
- The main editor the tools we will use unity 3d, we had previous experience with android studio but compared to unity there were some significant differences so finally we needed to deal with the user interface of unity engine.

3.8 Process Modeling

Process modeling is the process of creating a structural view of a system or process. In this process we will see that after the game is started which conditions are met by the user.

3.9 Functional requirements

This section includes all the functional requirements in our project in a sequential manner. It will help to describe the functionalities of our project. It means what our system or our project will do or the tasks included in it what will be their functionalities. It is all listed in the section of functional requirements. Such as:

3.9.1 Start

Input: Select the start game option from the menu.

Output: User will be able to play the game.

Process: Verification of the choice of game level will be made.

Description: It will be verified what game level the user has selected, and the game will start according to the user's choice.

3.9.2 Load

Input: Select the load game option from menu.

Output: User will be able to play the game that saved last time.

Process: Verification of the choice of game will be made.

Description: It will be verified what game level the user has selected, and the game will start according to the user's choice.

3.9.3 Setting

Input: Select the Setting game option from menu.

Output: User will be able to make game settings like sound on/off etc.

Process: Verification of the choice of game will be made.

Description: It will be verified what game setting the user has selected and the game will start according to the user's choice.

3.9.4 Exit

Input: Select the Exit game option from menu.

Output: User will be able to make game exit.

Process: Verification of the choice of game will be made.

Description: It will be verified what game setting the user has selected and the game will start according to the user's choice.

3.10 Non-Functional Requirements

Non-functional requirements are the characteristics or qualities of a system that describe how it should behave, rather than what it should do. These requirements typically define the constraints and limitations of a system and aspects that may affect its overall effectiveness and user satisfaction. The non-functional requirements of our project are listed below:

3.10.1 Cross-Platform Compatibility:

If applicable, specify the non-functional requirements for cross-platform play, ensuring that players on different operating systems can interact seamlessly.

3.10.2 Regulatory Compliance:

Ensure the game complies with relevant industry standards and regulations, especially if it involves gambling elements or other sensitive content.

3.10.3 Load Balancing:

Define load balancing requirements to evenly distribute player traffic among servers, ensuring optimal performance and resource utilization.

3.10.4 Security Updates and Patching:

Define requirements for releasing security updates and patches promptly in response to potential vulnerabilities or exploits.

3.10.5 Environmental Simulation:

If your game has realistic weather or environmental simulations, specify requirements for accuracy and consistency.

3.11 Use case modeling and Description

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases. It is a methodology used in system analysis to identify, clarify, and organize system requirements. Also finds out the reasons for achieving goals and the reasons for not achieving goals. The use case diagram for our game is given below:

3.12 System use case diagram for Wandering Spirit

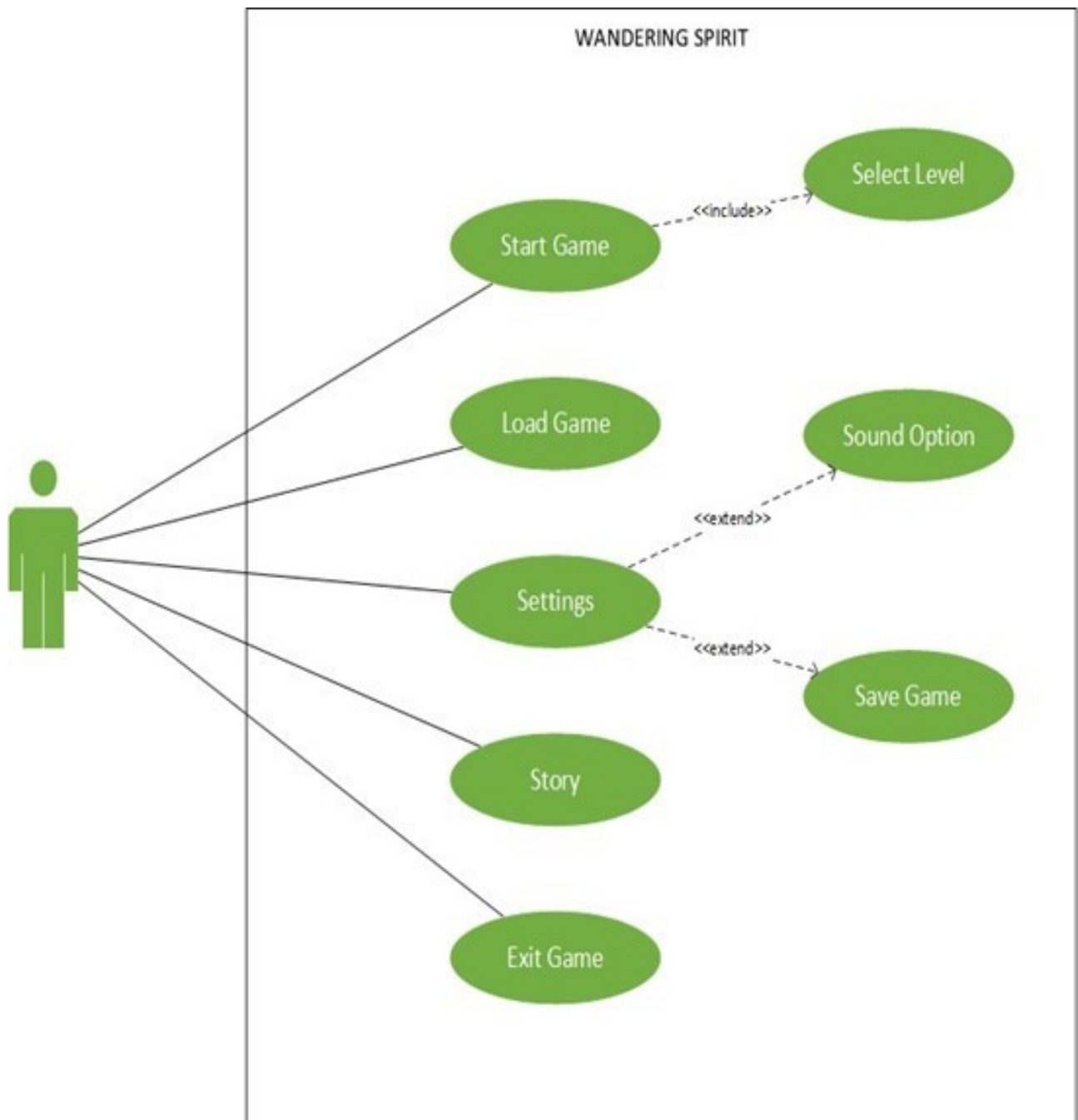


Figure 3.1: System use-case diagram

3.13 Use case description for all scenarios Wandering Spirit

3.13.1 Use case description for Start game

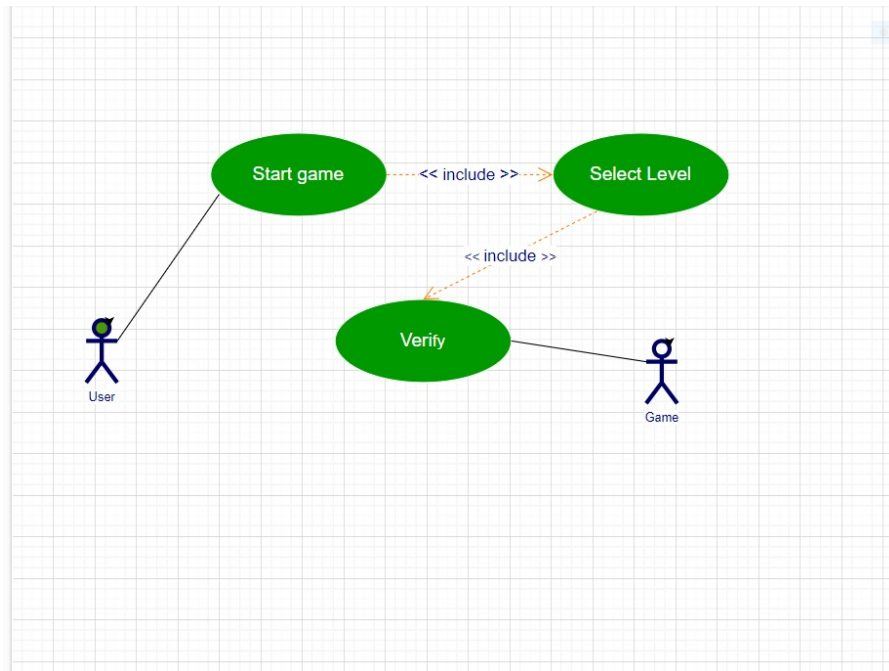


Figure 3.2: Start-game use-case diagram

3.13.2 Use case description for Load game

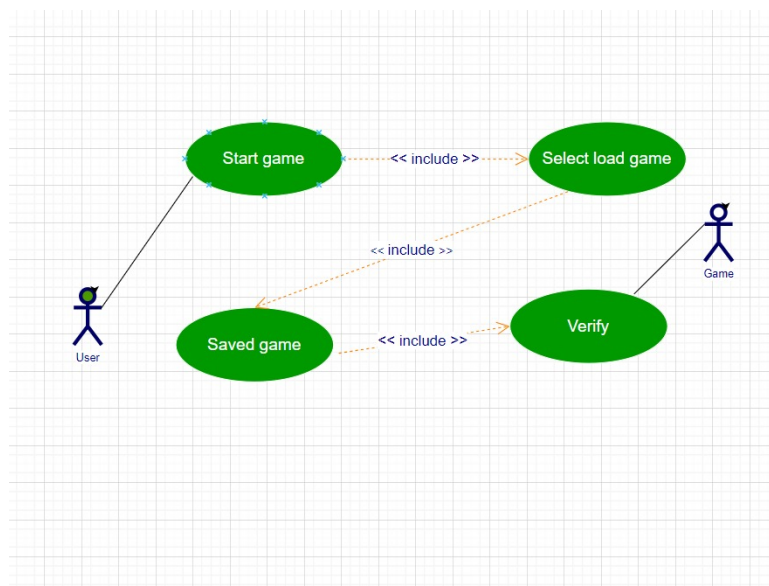


Figure 3.3: Load-game use-case diagram

3.13.3 Use case description for setting

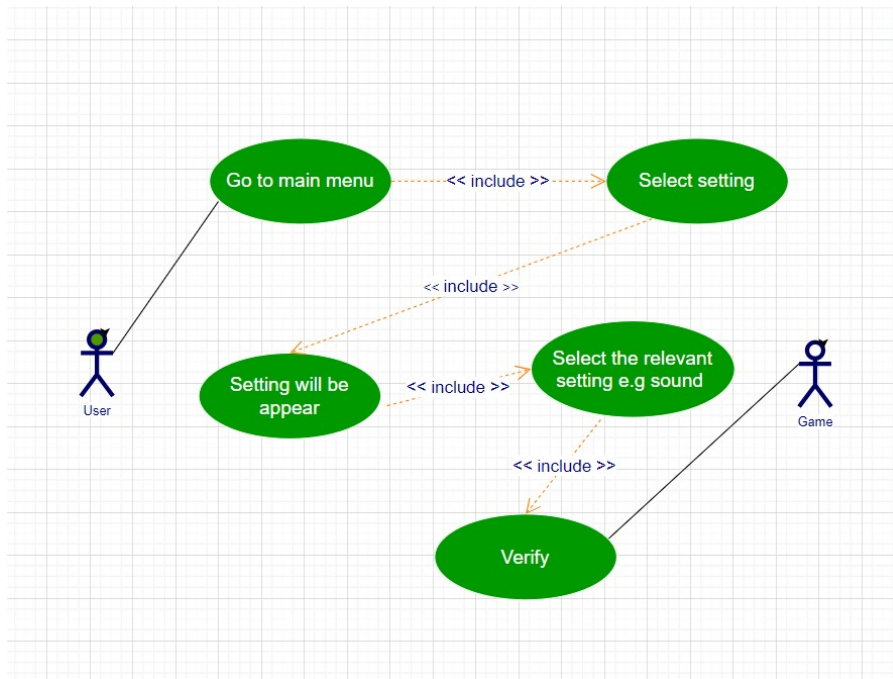


Figure 3.4: setting use-case diagram

3.13.4 Use case description for quit game

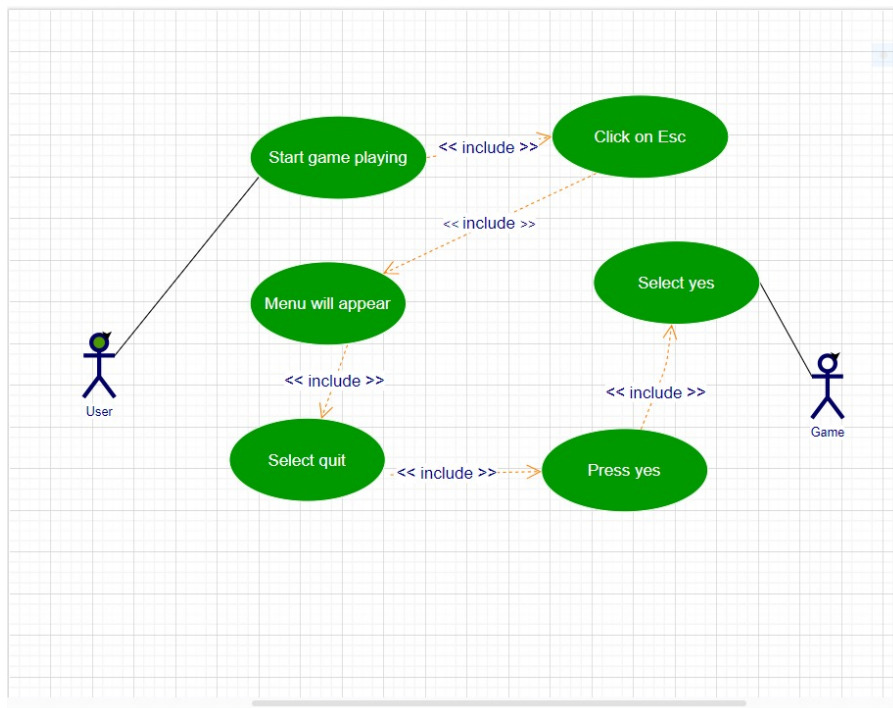


Figure 3.5: Quit-game use-case diagram

3.13.5 Start Game Use-Case Description

The "Start Game" use case in Fig 2.2 is essential for users to begin playing the game and requires them to navigate to the main menu, select a level, and initiate the loading of a new game on their system. Table shows the description of this Use-case.

Table 3.1: Use Case: Start Game Use-Case Description

Primary Actor	Users playing the game
Goal in Context	To start a new game by selecting a level
Preconditions	1. Game should be installed on the system while fulfilling hardware requirements. 2. The file has been triggered to run and the game screen will appear.
Triggers	The player needs to start a new game
Scenario	1. Go to the main menu of the game 2. Click start button 3. Select level 4. New game is loaded on the system
Priority	Essential, must be implemented

3.13.6 Load Game Use-Case Description

The "Load Game" use case in Fig enables users to continue playing from where they left off by loading a previously saved game. The use case requires the game to have been saved during the last session, and users must navigate to the main menu, click on the load game button, select the saved game, and initiate the loading process. This use case is essential for providing users with a seamless game play experience. Table shows the description of this Use-Case.

Table 3.2: Use Case: Load Game Use-Case Description

Primary Actor	Users playing the game
Goal in Context	To load the previous game
Preconditions	Game should be saved when last played
Triggers	Player needs to play from where they left the game
Scenario	1. Go to the main menu of the game 2. Click the load game button 3. Select the saved game 4. The game will be loaded
Priority	Essential, must be implemented

3.13.7 Setting Game Use-Case Description

The “Game Settings” use case in Fig allows users to customize their gameplay experience by changing game settings such as sound and music volume. The use case requires users to navigate to the main menu, click on the game settings button, and adjust the sound/music volume to their preference. This use case is essential for providing users with a personalized gameplay experience. Table shows the description of this Use-Case.

Table 3.3: Use Case: Setting Game Use-Case Description

Primary Actor	Users playing the game
Goal in Context	To change the game setting (sound)
Preconditions	Player is allowed to turn on or off sound/music of the game
Triggers	Player has a need to turn on or off sound/music of the game
Scenario	<ol style="list-style-type: none">1. Go to the main menu.2. Click on the game settings button.3. Click on Music/Sound and change the volume.4. Music or Sound Volume is changed.
Priority	Essential, must be implemented

3.13.8 Exit Game Use-Case Description

The “Exit Game” use case in Fig enables users to exit the game when they are done playing. The use case requires users to press the ESC key to access the game menu, click on the quit button, and confirm their decision to exit the game when prompted with a warning message. This use case is essential for providing users with a simple and straightforward way to end their gameplay session. Table shows the description of this Use-case.

Table 3.4: Use Case: Exit Use-case Description

Primary Actor	Users playing the game
Goal in Context	To exit from the game
Preconditions	A game is being played
Triggers	Player needs to exit from the game
Scenario	<ol style="list-style-type: none">1. Press ESC to access the game menu.2. Press the quit button.3. A warning message will be displayed.4. Select "yes".5. The game will be exited.
Priority	Essential, must be implemented

Chapter 4

Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. The following sections constitute this chapter:

4.1 System Architecture

The Architecture of this system is designed to be interactive and straightforward. It features a user-friendly interface that includes a main menu for user interaction. Additionally, there is an options menu that provides game instructions to assist players. An Exit option is available to allow players to quit the game at any time. The game itself is a third-person endless runner, where the player's character must run, dodge obstacles, and accumulate a score. Colliding with any obstacle results in the termination of the game.

- **The Game Loop:** This is the core of the game, which handles the game state, inputs, and updates the game world. It interacts with the Rendering and Physics Engine subsystems to provide the player with an immersive and engaging game experience.
- **Rendering:** This subsystem is responsible for rendering the game world, including the player character, obstacles, and environment. It takes inputs from the Game Loop and generates visual output that is displayed on the player's screen.
- **Physics Engine:** This subsystem simulates the physical laws of the game world, including gravity, collisions, and other forces. It takes inputs from the Game Loop and applies the appropriate physical effects to the game objects.
- **Display:** This is the final subsystem responsible for displaying the game output on the player's screen. It takes inputs from the Rendering subsystem and displays the visual output to the player.

4.1.1 Logical Viewpoint

The following is the architecture diagram of our proposed system in multiplayer mode.

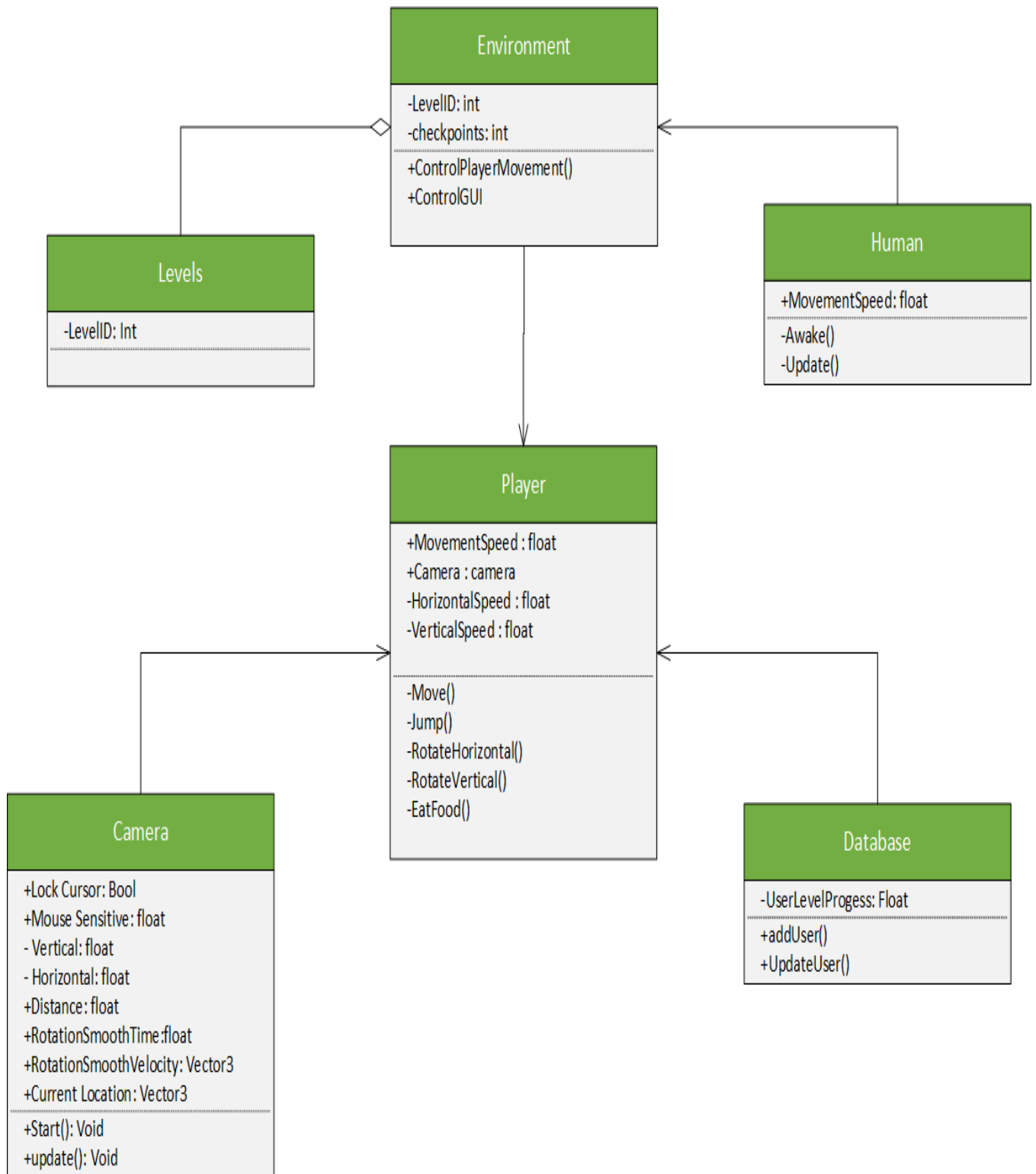


Figure 4.1: Logical Diagram

4.1.2 Explanation

- **The Game Loop:** This is the core of the game, which handles the game state, inputs, and updates the game world. In multiplayer mode, it interacts with the Multiplayer Game Logic subsystem to synchronize the game state between players.
- **Rendering:** This subsystem is responsible for rendering the game world, including the player character, obstacles, and environment. It takes inputs from the Game Loop and generates visual output that is displayed on the player's screen.
- **Physics Engine:** This subsystem simulates the physical laws of the game world, including gravity, collisions, and other forces. It takes inputs from the Game Loop and applies the appropriate physical effects to the game objects.
- **Input Handler:** This subsystem handles player input, such as touch screen inputs, keyboard inputs, or other game controller inputs. It takes inputs from the player and communicates them to the Game Loop.
- **Display:** This is the final subsystem responsible for displaying the game output on the player's screen. It takes inputs from the Rendering subsystem and displays the visual output to the player.

4.2 Sequence diagram

This subsystem handles network communication between players. It synchronizes the game state and updates between players, ensuring that all players have a consistent view of the game world Sequence Diagram

This is the sequence diagram of our proposed game. At the point when the player opens the application the main menu is shown, the time player starts the game, character starts running character must dodge obstacles and make a higher score as compared to the other player. Hitting any obstacle will cause the character to stumble and the game to end.

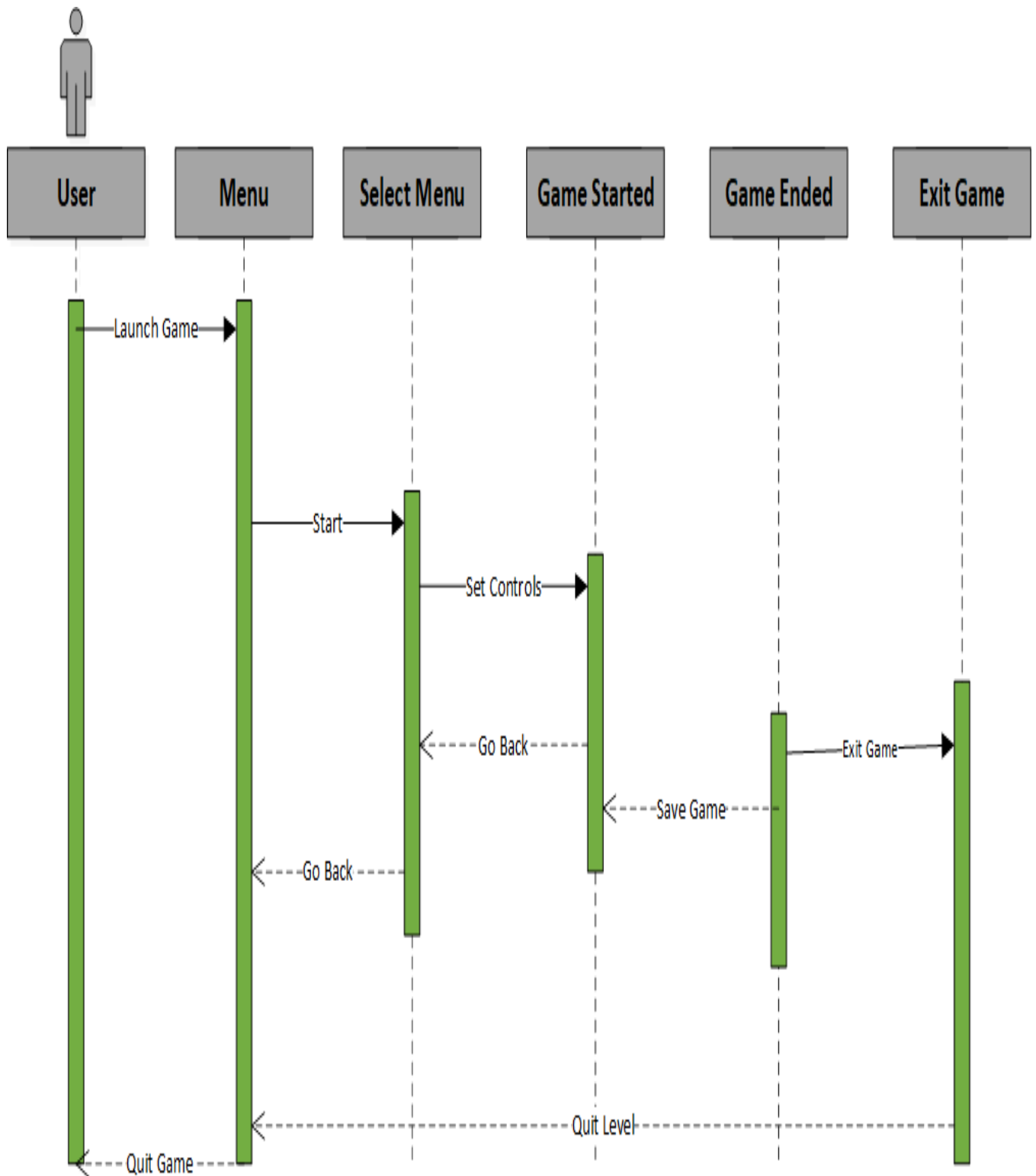


Figure 4.2: Sequence Diagram

4.3 Composite View

This System has four major parts to have a full package system, each part creates an abstraction for the multiple use cases along with their characteristics to represent the abstract view of the system and how the use cases are dependent on the system, and which use cases are required to work for the system fully. Fig 4.3 is a composite package diagram that portrays the dependence of each part in a fully working game environment.

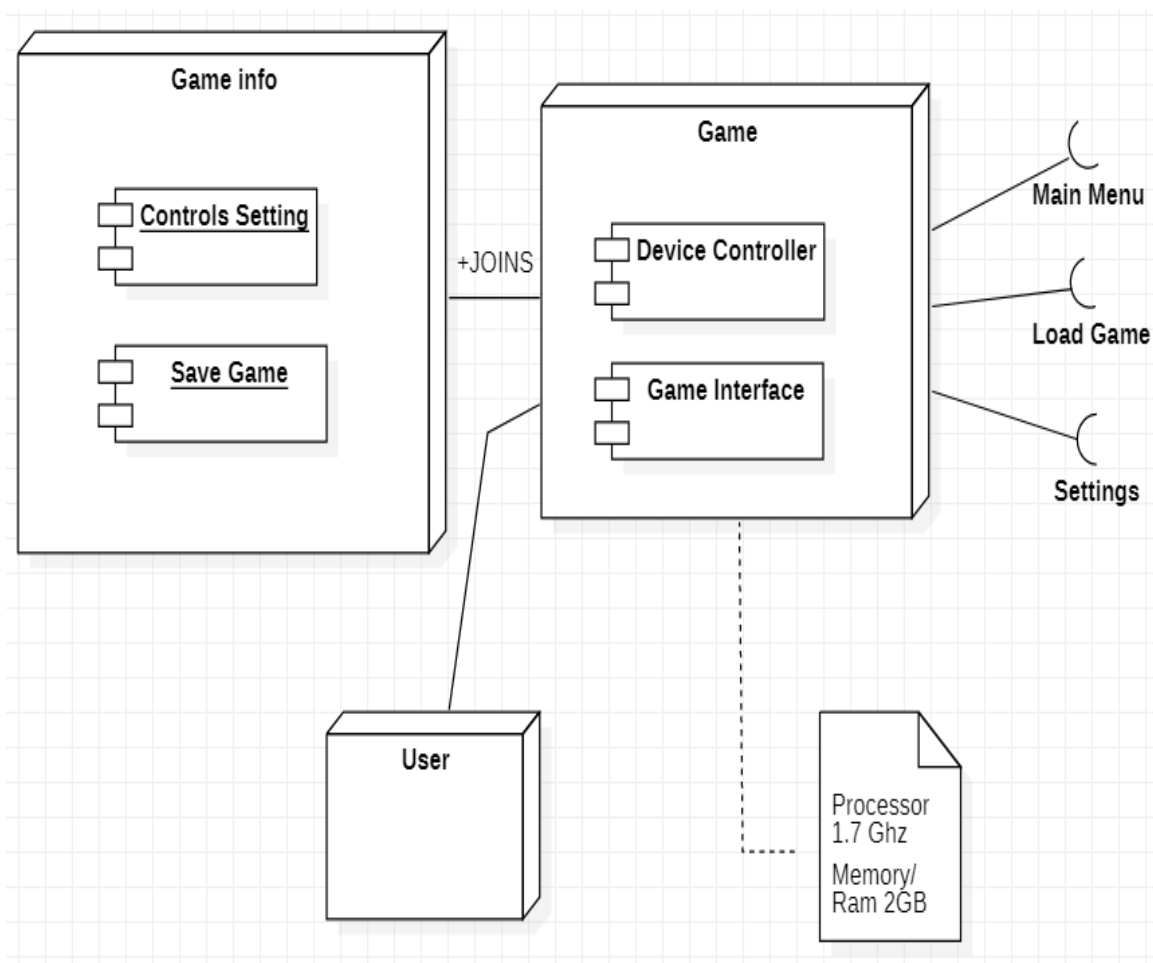


Figure 4.3: Composite View Diagram

4.4 Information Viewpoint

An information viewpoint is a graphical representation that depicts relationships among people, objects, places, concepts, or events within the system. The Information viewpoint shown in Fig. 4.4 provides a clear representation of the relationships between the entities in the game development system.

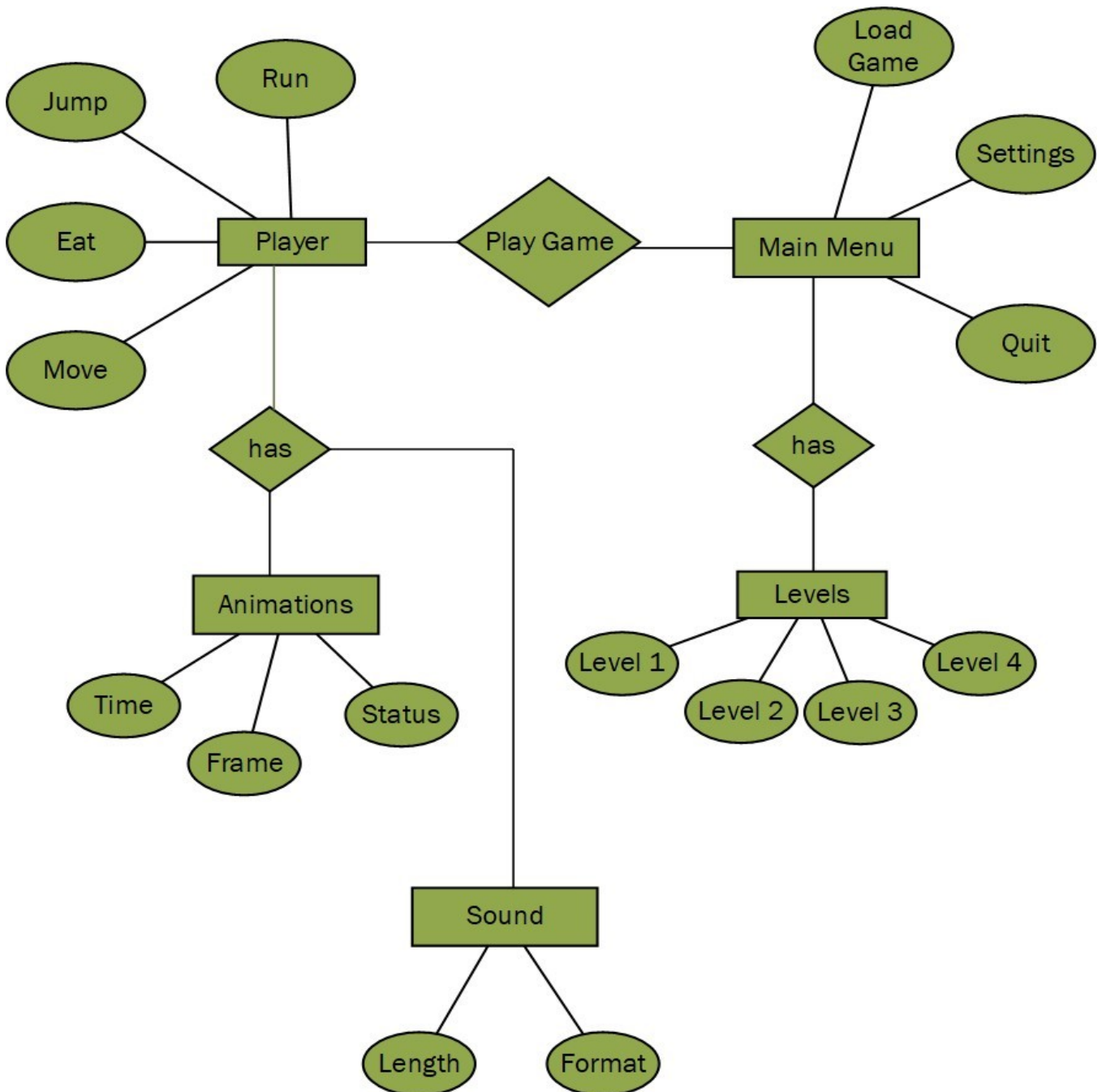


Figure 4.4: Information Viewpoint

4.5 State Dynamic view

A state machine diagram consists of states, transitions, events, and activities. State Machine diagrams are used to illustrate the dynamic view of a system. The State Machine Diagram in Fig 4.5 represents the various states that the game development system can be in and how it transitions from one state to another in response to various events or stimuli. The diagram can also show the actions and behaviors associated with each state and the conditions that must be met for a state transition to occur.

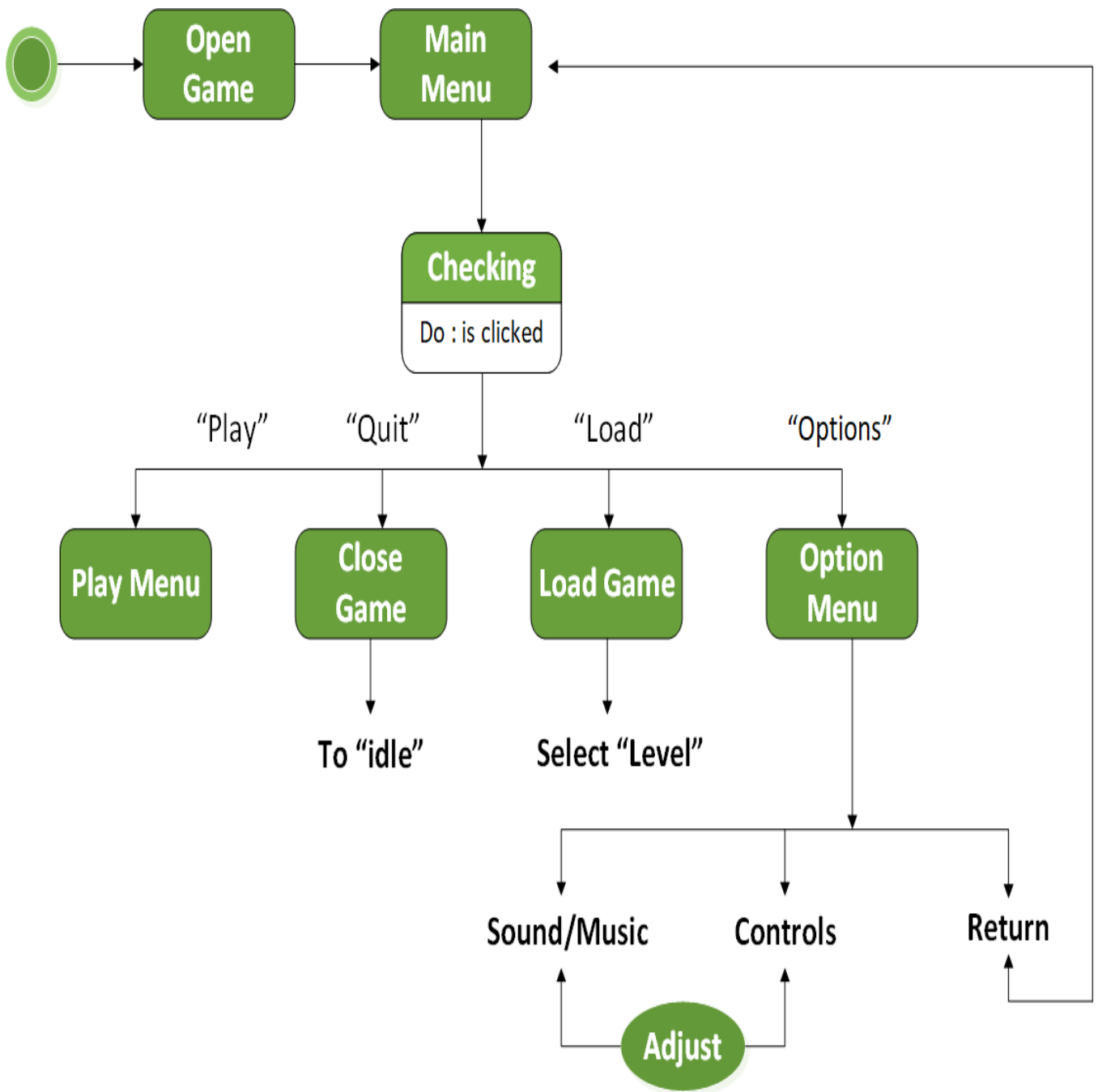


Figure 4.5: State dynamic view

4.6 Algorithmic Viewpoint

The Algorithmic Viewpoint section for the game development project focuses on the detailed design and implementation of the game's software, using Pseudo Code as the primary tool. It provides a clear and concise description of the various operations, methods, and functions used in the development process, aiding in troubleshooting and maintenance of the game's software.

Algorithm 1: Game Execution Algorithm

Data: Game state

Result: Game execution

```
1 while Game is running do
2   Start game;
3   Go to main menu;
4   if Start new game then
5     Open Level 1;
6     Click play game;
7   else
8     if Load Level then
9       Load previous Game;
10      Play Game;
11    else
12      if Player wants to pause then
13        Press Esc key;
14        if Player wants to save game then
15          Press Esc key then select save;
16        else
17          if Player wants to exit then
18            Press Esc key then exit from main menu;
19            Press close game;
20          else
21        else
```

Chapter 5

System Implementation

This chapter describes how our system is built, as well as the physical, logical, internal, and external components of the system, as well as the system execution, algorithm implementation, designs, and model. Each step of a system development is covered by system implementation. This chapter explains how design functional and non-functional components work together. The system's requirements from the previous two chapters have been implemented.

5.1 Software Requirements

- Visual Studio
- Blender
- Unity Engine

5.2 System Components

- Front end
- Back end
- UI Implementation

5.3 Front end

The meaning of front-end design is the visual part of a system. By which the user can interact with the system. While developing the game we tried to make it user-friendly and we intended to keep it as simple and creative as possible. From the point of view of our project front-end refers to the game environment design, character building, design of the path along which the character will run and all other visual graphics that come along with it. That means we considered non coding part as our front-end design.

5.4 Game Environment in unity

Since we made our environment based on road and added roadside obstacles, first we imported the file on unity. Then we fixed our road position as needed. With the help of different types of brushes on unity we designed the environment in the whole terrain. The figure shows the game's view of how our game's running environment will look like. After that we updated this by adding some textures of road and other related materials.

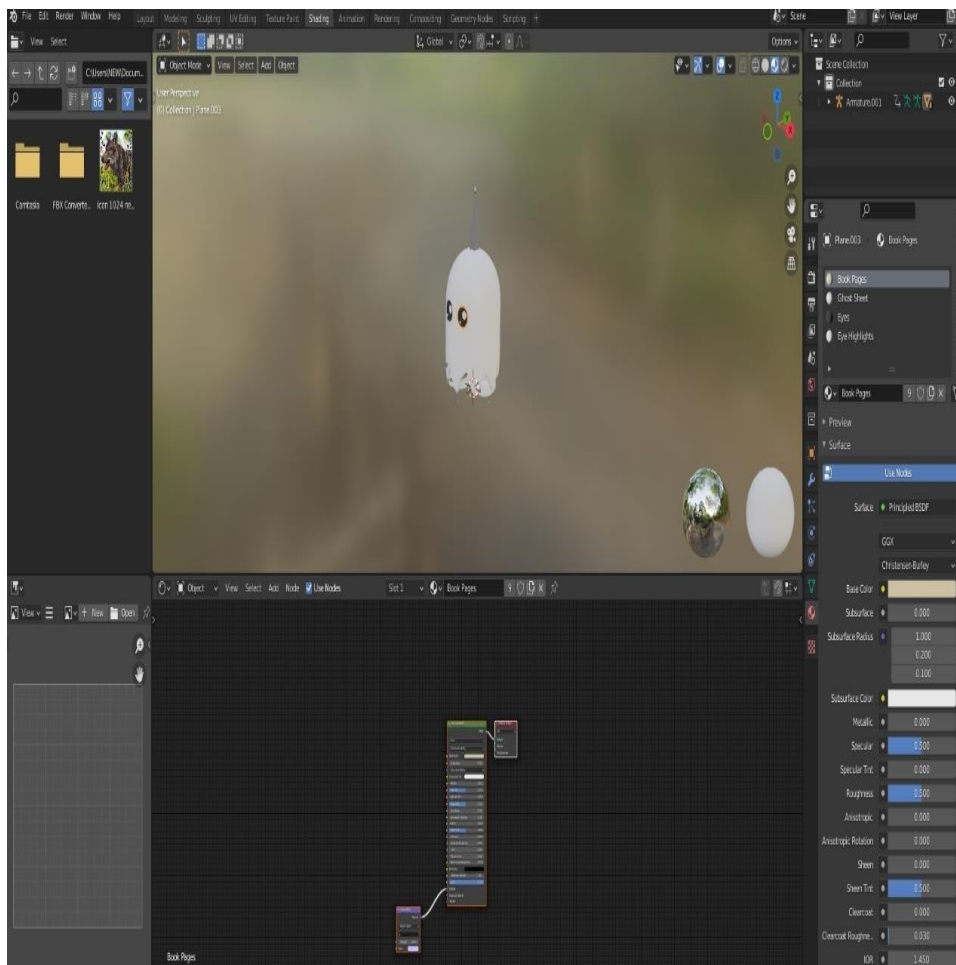


Figure 5.1: Game Environment

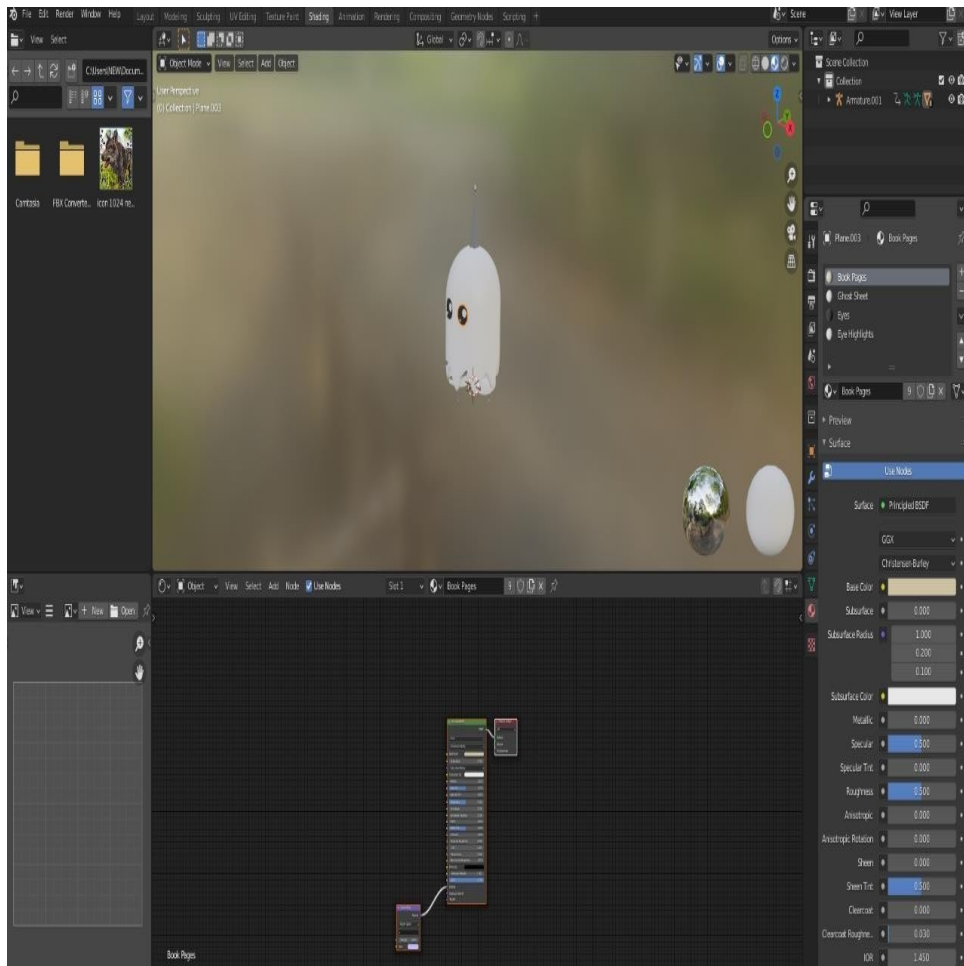


Figure 5.2: Game Environment

5.5 Other Options

The following are some of the other modules of our project. These modules provide the user with more options regarding their game play and experience.

The following are some of the other modules of our project. These modules provide the user with more options regarding their game play and experience.



Figure 5.3: Character Interface

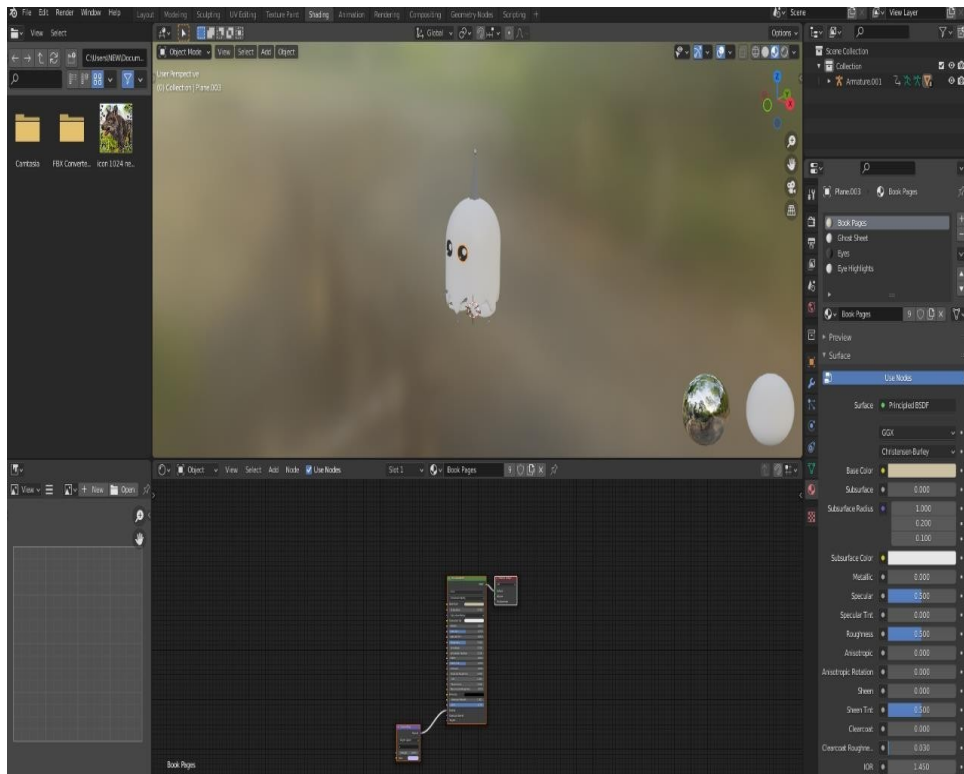


Figure 5.4: Character Side Interface + Animation



Figure 5.5: Man Character Model



Figure 5.6: Animal Character Model

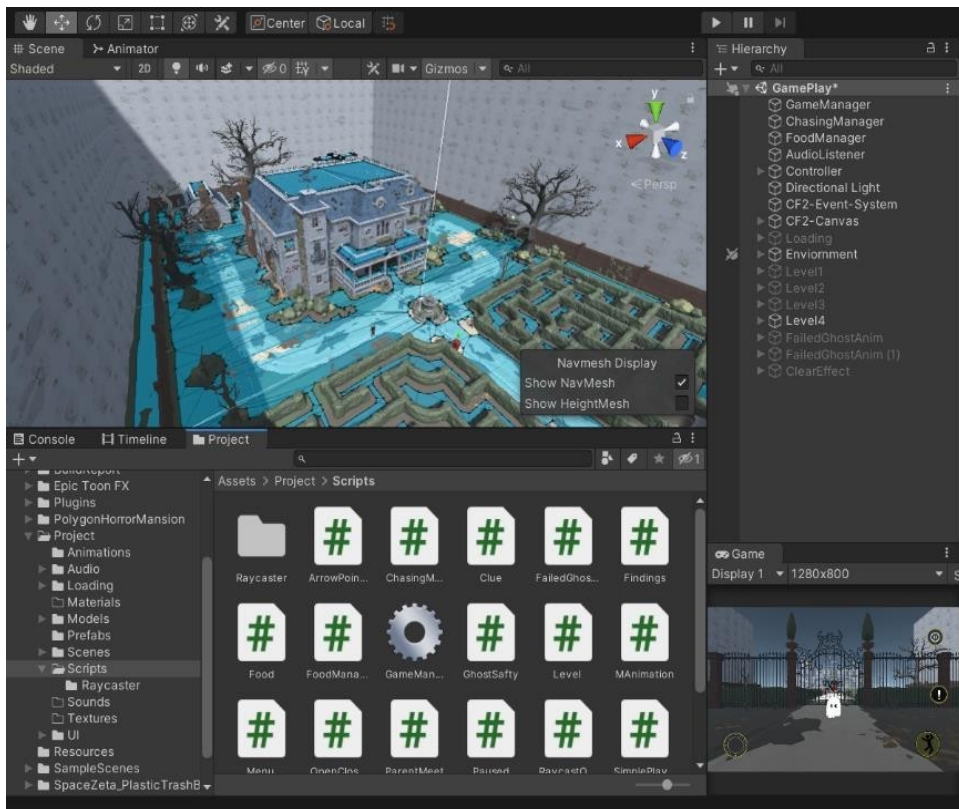


Figure 5.7: House Asset Front

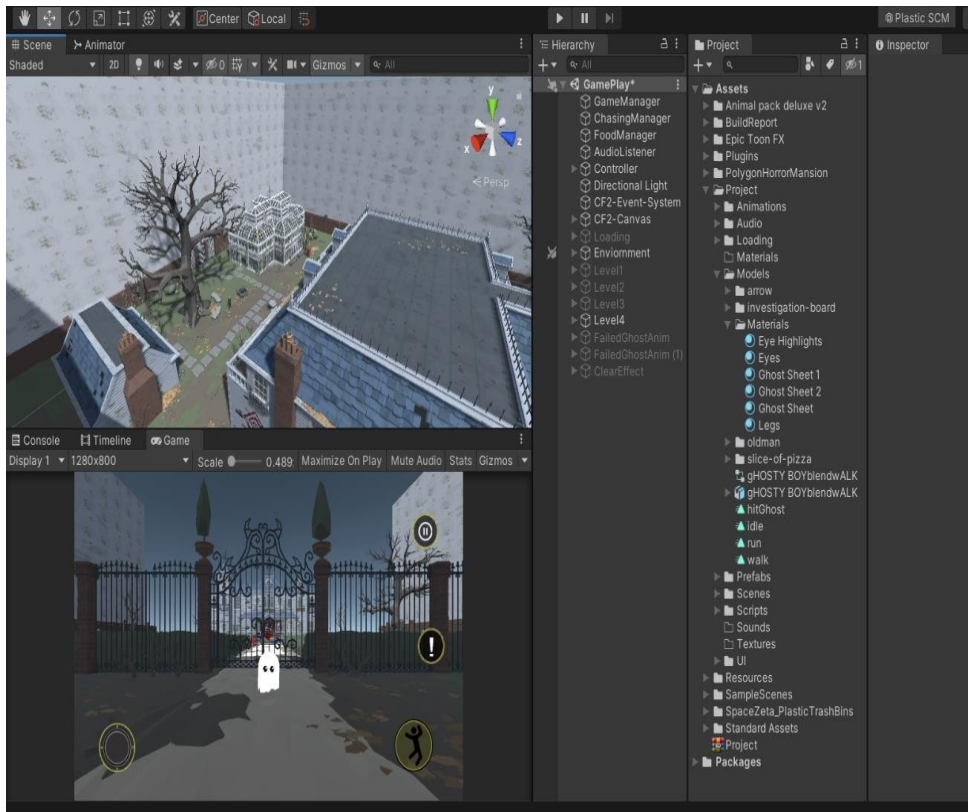


Figure 5.8: House Asset Side Interface

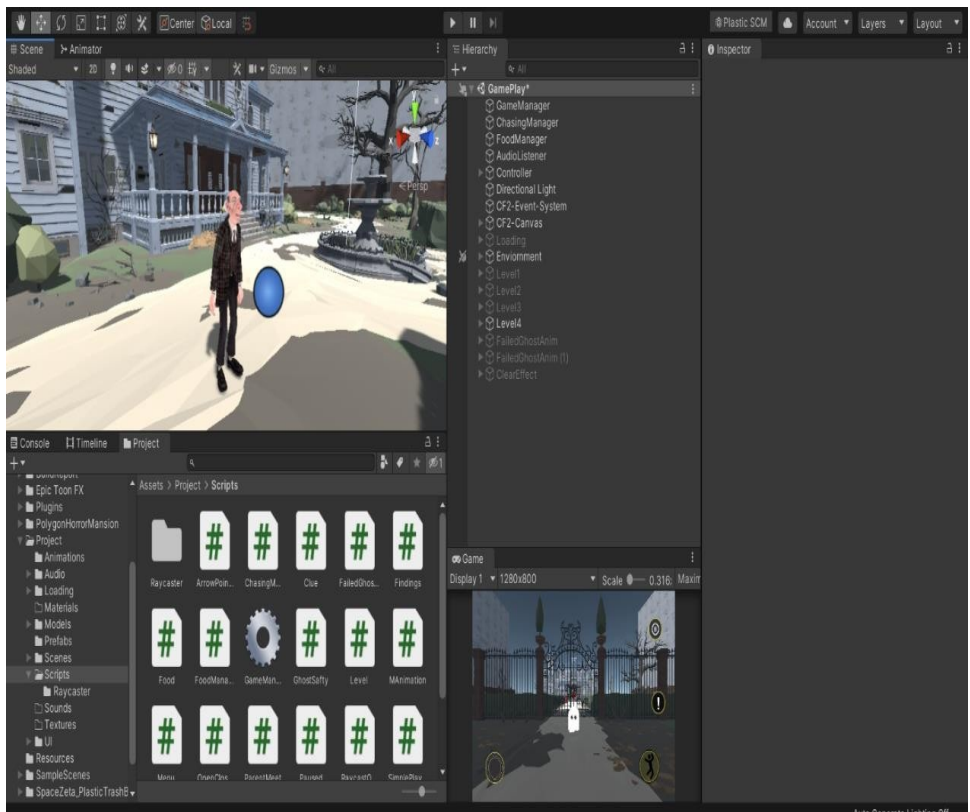


Figure 5.9: game Scripts

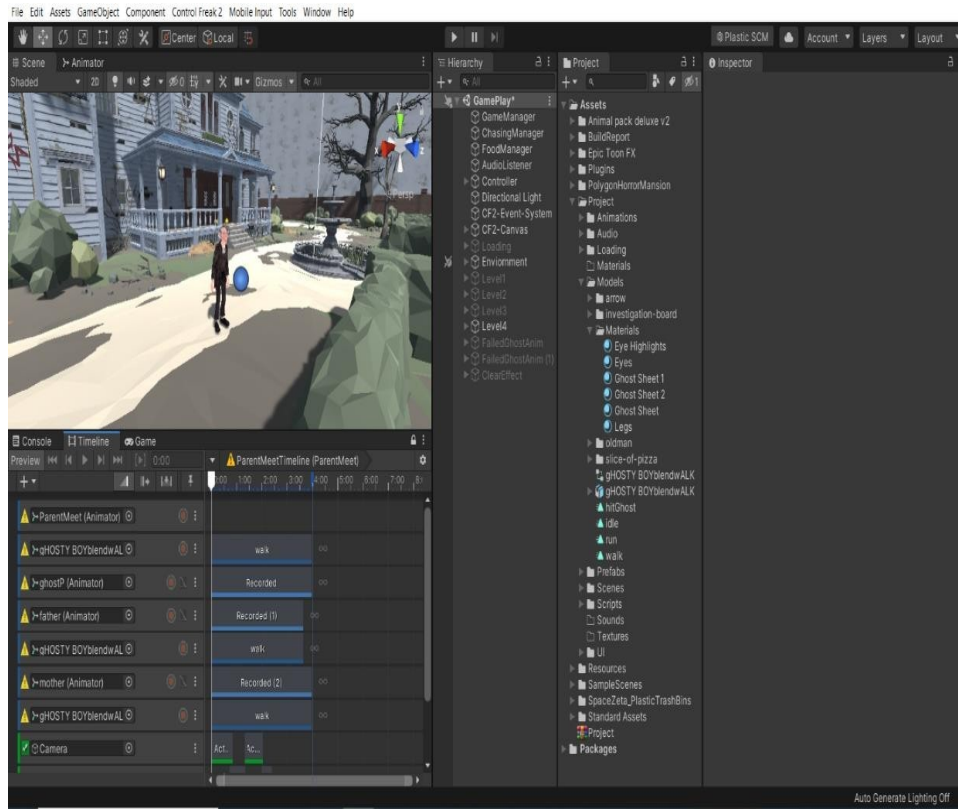


Figure 5.10: Man Character Animations

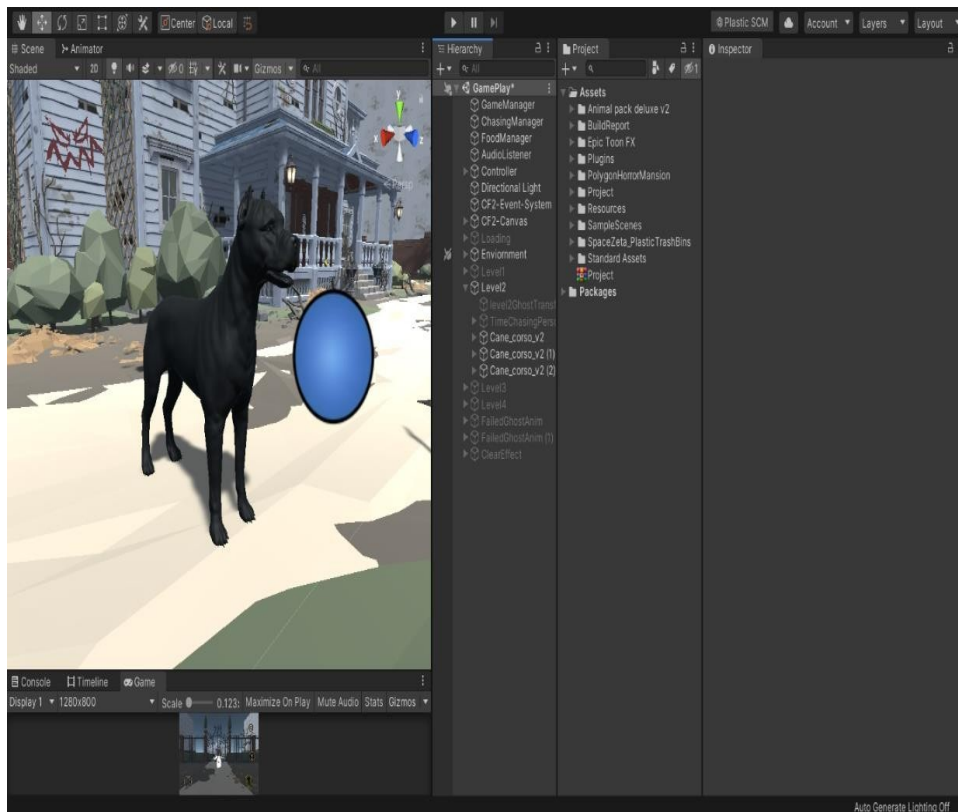


Figure 5.11: dog in environment



Figure 5.12: Introduction Video of Game

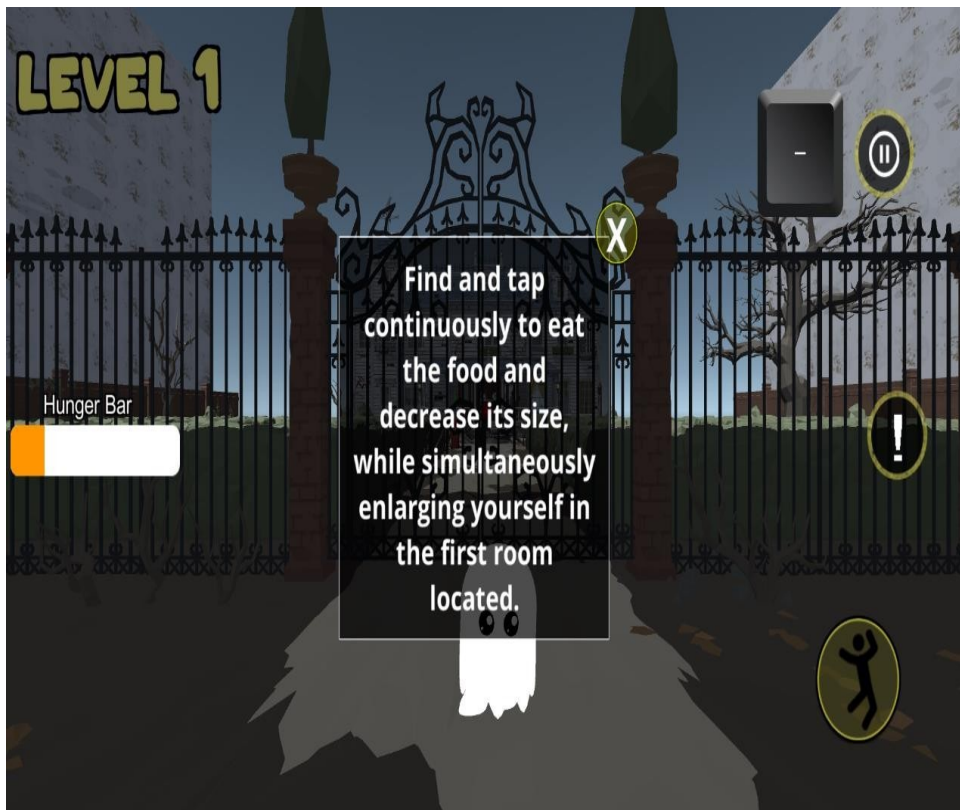


Figure 5.13: 1st Level Interface



Figure 5.14: Eating food



Figure 5.15: Character Vanished in Trash-Bin From Old Man



Figure 5.16: Character Running

Chapter 6

System Testing and Evaluation

6.1 Chapter Overview

Testing is an essential part of software development that helps to identify and correct errors in a program. In the context of game development, testing plays a critical role in ensuring that the game meets the expectations of players and functions as intended. This chapter focuses on the testing of our project which is an endless runner game, endless runner is a popular genre of mobile games that involves controlling a character to run endlessly and avoid obstacles to score points. The objective of this chapter is to describe the testing methodologies, tools, and techniques used in the development of our project, and to evaluate the effectiveness of these methods in identifying and resolving issues. This chapter also discusses the challenges and limitations of testing an endless runner game and provides recommendations for improving the testing process. By the end of this chapter, the reader will have a comprehensive understanding of the testing process for an endless runner game and the importance of testing in game development.

6.2 Testing and Methodologies and Techniques

This describes the ways in which our project was tested. Below is a list of all techniques used for our projects testing in detail.

- **Manual Testing:** This involves testing the game manually by playing it and observing its behavior. It is time-consuming but the most successful process, requiring testers to play the game multiple times to identify bugs and glitches. Testers can use a variety of devices to test the game, including mobile phones, tablets, and emulators. Most of the testing of our project is done through manual testing.
- **Functional Testing:** This involves testing the game's functionality, such as the movement of the character, collision detection, and scoring system. We created different test cases to keep track of all the progress we made and to ensure that our project is working properly.

6.3 Test Cases

6.3.1 Character Movement Test Case

This figure 6.1 describes that When the user plays the game, they will be able to move freely on the map.

Table 6.1: Test Case TC-1: Character Movement Test Case

Test ID	TC-1
Test Name	Character Movement
Date of Test	11/09/2023
Name of Application	Wandering Spirit
Description	When the user plays the game, they will be able to move freely on the map.
Input	Start Game
Expected Output	Player will be able to move
Actual Output	Player is moving
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.2 3rd Person User Perspective Test Case

This figure 6.3 describes that When the user plays the game, they will be able to use 3rd Person User Perspective, which means they are able to see the character in front of the camera.

Table 6.2: Test Case TC-2: 3rd Person User Perspective Test Case

Test ID	TC-2
Test Name	3rd Person User Perspective
Date of Test	15/09/2023
Name of Application	Wandering Spirit
Description	When the user plays the game, they will be able to use 3rd Person User Perspective, which means they are able to see the character in front of the camera.
Input	Start Game
Expected Output	Player Character will not be behind the camera.
Actual Output	Player Character is not behind the camera.
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.3 Eating Ability Test Case

This figure 6.3 describes that When the user plays the game, they will be able to eat the food.

Table 6.3: Test Case TC-3: Eating Ability of Ghost Test Case

Test ID	TC-3
Test Name	Eating Ability of Ghost
Date of Test	17/09/2023
Name of Application	Wandering Spirit
Description	When the user plays the game, they will be able to eat the food.
Input	Start Game
Expected Output	Ghost will eat the food
Actual Output	Ghost eats the food
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.4 Vanish ability

This figure 6.4 describes that When the user plays the game, they will be able to vanish from different characters by hiding themselves in Trash-bins.

Table 6.4: Test Case TC-4: Vanish Ability Test-case

Test ID	TC-4
Test Name	Vanish Ability of Ghost
Date of Test	19/09/2023
Name of Application	Wandering Spirit
Description	When the user plays the game, they will be able to vanish from different characters by hiding themselves in Trash-bins.
Input	Start Game
Expected Output	Ghost will vanish by pressing a button
Actual Output	Ghost successfully vanishes in Trash-bins
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.5 Select Option from Menu Test Case

This figure 6.5 describes that the game should have a graphical user interface with a menu from which the user can select settings related to control, exit, and start the game.

Table 6.5: Test Case TC-5: Select Option from Menu TestCase

Test ID	TC-5
Test Name	Select Option from Menu
Date of Test	21/09/2023
Name of Application	Wandering Spirit
Description	The game should have a graphical user interface with a menu from which the user can select settings related to control, exit, and start the game.
Input	Select Option
Expected Output	Options will be displayed
Actual Output	Options are displayed
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.6 New Game Test Case

Test ID	TC-6
Test Name	New Game Option
Date of Test	21/09/2023
Name of Application	Wandering Spirit
Description	If a player wants to start the game from the beginning, they should be able to start a new game and have the option to start the new game.
Input	Click the new button
Expected Output	A loading screen will appear, and then the game will start from the beginning.
Actual Output	A loading screen appears, and the game starts.
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.7 Pause Option

Table 6.6: Test Case TC-7: Pause Option

Test ID	TC-7
Test Name	Pause Option
Date of Test	23/09/2023
Name of Application	Wandering Spirit
Description	There should be a pause option in the game. If the player wants to pause the game for a while, they should have a pause option.
Input	Click Pause Button
Expected Output	The game will pause
Actual Output	The game is paused
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.8 Exit Option Test Case

Table 6.7: Test Case TC-8: Exit Option TestCase

Test ID	TC-8
Test Name	Exit Option
Date of Test	23/09/2023
Name of Application	Wandering Spirit
Description	The game should have a quit option. If you don't want to play the game, press the quit button to exit the game at any time.
Input	Click Exit Button
Expected Output	Exit from Game
Actual Output	Exit from Game
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.9 Hurdles Test Case

Table 6.8: Test Case TC-9: Hurdles Test Case

Test ID	TC-9
Test Name	Hurdles for Players
Date of Test	25/09/2023
Name of Application	Wandering Spirit
Description	The game should have hurdles that make the game intense for players.
Input	Player will run when dogs run towards him
Expected Output	Hurdle will be crossed
Actual Output	Hurdle is crossed
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.10 Game should have hurdles for players

Table 6.9: Test Case TC-10: Hurdles Test case(2)

Test ID	TC-10
Test Name	Hurdles for Players
Date of Test	27/09/2023
Name of Application	Wandering Spirit
Description	The game should have hurdles that make the game intense for the players. If the player does not run from running dogs coming towards him, then the player will die, and the level will be failed.
Input	Player does not run from running dogs
Expected Output	Player dies, and the level fails
Actual Output	Player dies, and the level fails
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.11 Player must complete the level

Table 6.10: Test Case TC-11: Completing a Level (Failure Condition)

Test ID	TC-11
Test Name	Completing a Level
Date of Test	29/09/2023
Name of Application	Wandering Spirit
Description	To unlock the next level, the player must complete the level. If the player does not complete the level, then the player should not be promoted to the next level.
Input	Player does not complete the level
Expected Output	Player is not promoted to the next level
Actual Output	Player is not promoted to the next level
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.12 Detecting ghost near Old-man character

Table 6.11: Test Case TC-12: Old Man Character Test Case

Test ID	TC-12
Test Name	Old Man Character
Date of Test	01/10/2023
Name of Application	Wandering Spirit
Description	The old man wants to catch the ghost if he sees the ghost in the environment.
Input	Play Game
Expected Output	Ghost is not caught
Actual Output	Ghost is not caught
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.13 Detecting ghost near Old-man character

Table 6.12: Test Case TC-13: Old Man Character test Case(2)

Test ID	TC-13
Test Name	Old Man Character
Date of Test	04/10/2023
Name of Application	Wandering Spirit
Description	The old man wants to catch the ghost if he sees the ghost in the environment. If the ghost is caught, then the level will fail.
Input	Play Game
Expected Output	Ghost is caught (Level Failed)
Actual Output	Ghost is caught (Level Failed)
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.14 Player should be able to change the volume level of game

This figure 6.13 describes that there should be a sound button in the option screen from where the user can change the volume level of the game.

Table 6.13: Test Case TC-14: change volume of game

Test ID	TC-14
Test Name	Adjusting Sound Volume
Date of Test	7/10/2023
Name of Application	Wandering Spirit
Description	There should be a sound button in the option screen from where the user can change the volume level of the game.
Input	Click on the Sound Button (Settings)
Expected Output	Adjust Sound Screen will be displayed
Actual Output	Adjust sound screen is displayed
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.3.15 Game should be in 3D environment

This figure 6.14 describes that When the user plays the game, they should be able to see a 3D environment.

Table 6.14: Test Case TC-15: Game should be in 3D environment

Test ID	TC-15
Test Name	3D Environment Display
Date of Test	10/10/2023
Name of Application	Wandering Spirit
Description	When the user plays the game, they should be able to see a 3D environment.
Input	Start Game
Expected Output	User should be able to see a 3D environment
Actual Output	User can see a 3D environment
Test Role (Actor)	Sanawar Iqbal
Test Verified by	Mahpara Zamid

6.4 User Manual

Every device comes with a precise user guide. It is basically the set of rules and interactions created by a development team of the system. To catch the right customers and the right usage of system. There are also some technical issues to cover that we will mention in our user manual. This is a universal document that is made on behalf of every programmer or non-programmer user of the system to get maximum benefit out of product.

6.4.1 Introduction

The User Manual contains all the essential information for the user to make full use of the information system. This manual includes a description of the system functions and capabilities, contingencies and alternate modes of operation, and step-by-step procedures for system access and use. Use graphics where possible in this manual. The manual format may be altered if another format is more suitable for the project.

6.4.2 Hardware and software requirement for system

Here are the game system requirements (minimum):

- Platform: Windows 7 and above
- RAM:4 GB
- Memory: 1.5-2 Gb
- Graphics: Default graphics of system

6.4.3 Installation Guide for Application

This is based on platform of PC so the user can extract game files from the archived file of Game and after game is extracted then User (player) can open the game from the game file extracted and start the game through Game.exe Application to enjoy the game.

6.4.4 Operational Manual

- First run game.exe file
- Main Menu Screen Will Appear



Figure 6.1: Game main menu

- By clicking the new game button, the load screen will appear then the game will start.
- By Clicking load game button, save game will be start.
- By Clicking the Setting Button, setting menu will appear.
- Setting Screen is appeared
- By clicking the exit button, you can exit.



Figure 6.2: Main Menu New Game Button



Figure 6.3: Main Menu Load Screen Button



Figure 6.4: Main Menu Setting Button



Figure 6.5: Setting Screen



Figure 6.6: Main Menu Exit Button



Figure 6.7: Ending interface



Figure 6.8: Story button interface

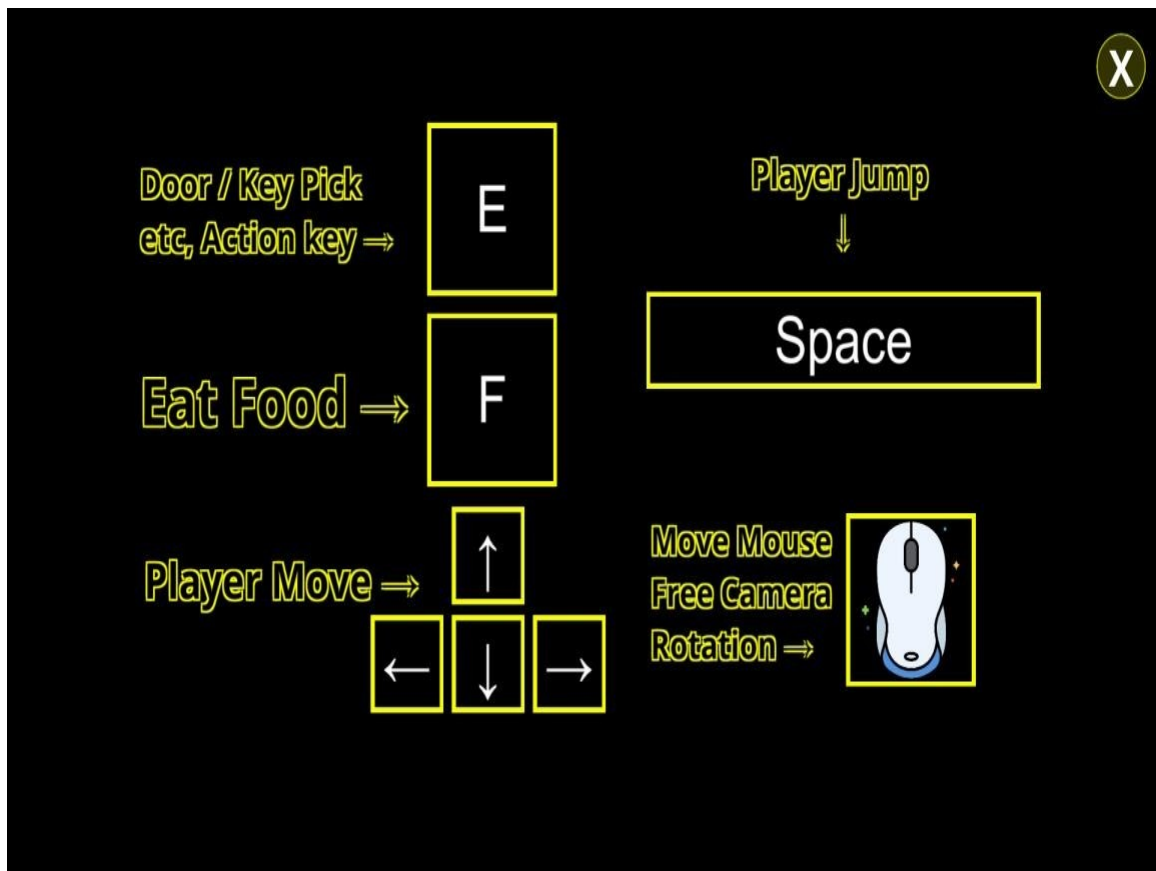


Figure 6.9: user controller guide

Chapter 7

Conclusions

Wandering spirit is a web-based game[9]. The “Wandering Spirit” is a complete strategy game with different levels. The main character of this game is a little ghost who loses its parents in a human neighborhood. The baby is afraid of people and has not adopted many ghost tricks yet. Therefore, it is difficult for it to return to its parents. Now it must find food and keep itself hidden in a crowded town. There are four levels, and, in each level, the gamer must hide the ghost from people and feed it. Our game will serve to entertain, relax, and add excitement to the players. This game may also increase the IQ level of the users in the game and in real life as it is based on Survival Techniques. It is a single player game and has the player move through different stages that progress the story. The main goal of the player is to use some strategies and logic to remain hidden (ghost) from the sight of people.

Introduces mental health issues through gaming medium. In our game will have a free-to-play environment having a vast parameter of survival options, as well as varied strategies and logic to save the ghost and survive from other people. We saw that in the traditional endless runner game there was something missing. Users would play the game for quite a while and get tired of it. Seeing the same thing and following a similar way the human brain gets fed-up of it. It needs energy to remain on track and not get exhausted. Our game likewise gives users numerous degrees of game and the subject of Casper animation. This project aimed to target the gaming industry, especially strategy games. Our game will serve to entertain, relax, and add excitement to the players. This game may also increase the IQ level of the users in the game and in real life as it based on Survival Techniques The idea behind this project is to provide people with a unique way and unique experience playing the same endless runner game that have been in the market for years and years (for example temple run etc.). Our aim is to make a game that will be more exciting for the users to play and pass their time. Provides an immersive survival experience. Our game focuses on the animation, story line and create a game mode that anyone could play.

7.1 Future Work

The future of a desktop survival game offers many ways to make it better. We can add new places to explore, new things to find, and make crafting more interesting. The game’s characters and creatures can also be smarter. Players can work together or compete against each other in multiplayer modes. We can make the game more exciting with a good story and missions. The game will also look and sound better with improved graphics and sounds. Making it easier to use and customizing it for

players will keep them happy. In future we will try to make this project available on multiple IOS like mac android etc. All these things will help keep players interested and make the game successful in the long run.

References

- [1] [https://casper.fandom.com/wiki/Casper_\(film_franchise\)#Casper:_A_Spirited_Beginning_\(1997\)](https://casper.fandom.com/wiki/Casper_(film_franchise)#Casper:_A_Spirited_Beginning_(1997))
Cited on p. 2.
- [2] <https://ieeexplore.ieee.org/document/6295169> Cited on p. 2.
- [3] <https://ieeexplore.ieee.org/document/6295169> Cited on p. 5.
- [4] [https://en.wikipedia.org/wiki/Rust_\(video_game\)](https://en.wikipedia.org/wiki/Rust_(video_game)) Cited on p. 6.
- [5] https://rust.fandom.com/wiki/Research_Paper Cited on p. 6.
- [6] <https://gamefaqs.gamespot.com/boards/929197-resident-evil-5/48308015> Cited on p. 7.
- [7] <https://en.wikipedia.org/wiki/Subnautica> Cited on p. 7.
- [8] [https://en.wikipedia.org/wiki/Green_Hell_\(video_game\)](https://en.wikipedia.org/wiki/Green_Hell_(video_game)) Cited on p. 8.
- [9] https://en.wikipedia.org/wiki/Casper_the_Friendly_Ghost Cited on p. 52.