TALHA SHAUKET
01-235192-082
FAWAD WAHEED
01-235192-095

Race 'Em Multiplayer Game

Bachelor of Science in Information Technology

Supervisor: Zubaria Inayat

Department of Computer Science
Bahria University, Islamabad

June 2023

Certificate

We accept the work contained in the report titled **Race 'Em Multiplayer Game**, written by Mr. Talha Shauket and Mr. Fawad Waheed as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Information Technology.

Approved by . . . :

Supervisor: Zubaria Inayat
Internal Examiner:

_____

External Examiner:

_____

Project Coordinator: Ms. Zubaria Inayat (Sr. Lecturer)

_____

Head of the Department: Dr. Arif ur Rahman (Sr. Associate Professor)

_____

June 9^{th}, 2023

Abstract

An arcade game of endless runner category in which there are two different playing modes. These modes are offline single player and online multiplayer to keep the user busy and interacted with the game. Along with providing the user with the option to choose from a range of avatars, in-game shop where users can spend the coin they collect as well as change the environment in which they play to keep the user interacted and keep then interested.

Acknowledgments

In the Name of Allah, the Merciful, the Beneficent. Praise be to the Lord of all worlds. Prayers and peace be upon our Prophet, Muhammad, his family and all of his companions. I would like to acknowledge and give my warmest thanks to my supervisor Zubariya Inayat who made this work possible. Her guidance and advice carried me through all the stages of writing my project. I would also like to thank my university faculty for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

TALHA SHAUKET
FAWAD WAHEED
Islamabad, Pakistan

June 2023

"Mastering others is strength. Mastering yourself is true power."

Lao Tzu

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

SPM	Single Player Mode
MPG	Multiplayer game
UE	Unity Engine
SE	Software Engineering
MPM	Multiplayer Mode
SQL	Structured Query Language
XML	Extensible Markup Language

Chapter 1

Introduction

This chapter is the introduction to the listing of some key game specifications for our final year project and some target game mechanics. By reviewing a few models that describe these game mechanics you'll have good idea of how they work. Lastly we'll review the character control scheme for the game. Endless runner games tend to provide a sense of relaxation and freedom to the user but to make this game there are a large number of challenges involved and require a large amount of game content. Our project is a game named Race 'Em. In this game users from anywhere around the world can download and compete that is something previously absent from the endless runner games. We took that opportunity to make a game that would be both unique in its category as well as provide the users with some sense of familiarity so that users would be attracted towards our game. The game will have both types you can either play it in single player mode like all the other games of the category or you can switch to multiplayer and compete with players around the world.

1.1 Background

Technology plays a vital role on our day to day life style. There are lots of various gaming applications that are being used daily for entertainment purposes. This project is a player versus player or player versus computer game based on what mode you want to play it in. Therefore, this is an entertainment project that provides a splendid experience, designed in a way that provides challenging stat for leveling up and competition. This bachelors' thesis provides the theoretical aspects of multiplayer endless runner game physics and how they can be applied to the unity game engine.

1.2 Objectives

The main objectives of this project are as follows.

- To develop an endless runner game with different playing modes
- To provide an interactive and challenging gameplay experience
- To expand our knowledge and skillset

1.3 Problem description

We saw that in traditional endless runner game there was something missing. Users would play the game for some time and get fed-up of it. Seeing the same thing and following the same path the human mind tends to get bored. It needs excitement in order to stay focused and not get bored. This is where we thought there are no multiplayer endless runner games in the market. This is where "Race 'em" comes in. It will provide the user with opportunity to compete with other players in online multiplayer and see which player is better at the game.

1.4 Methodology

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

1.4.1 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 allows for a faster development cycle and can result in a higher quality product due to continuous testing and evaluation throughout the development process.

1.5 Features

Main features of the game are

- Two different playing modes i.e. single player and multiplayer

- Background scenery can be changed
- Power ups such as score multiplier, coin multiplier etc
- Coins collected during run will be stored in wallet which can be used to purchase items
- As you cover more distance the speed of the character will increase and hence the difficulty level will also increase
- Data will be saved in local database

1.6 Project Scope

1.6.1 Introduction

"Race 'Em" is a mobile application-based game that will allow multiple users to play the game from anywhere in the world. There are two playing modes the offline mode can be played by a single anywhere anytime without the need to connect with the internet. In multiplayer mode user will connect with internet and compete with other players there will be minimum two players that can play in multiplayer mode. The game will be available on android OS in beginning.

1.6.2 Goals

By doing this project we are going to be able to:

- To develop practical skills such as problem-solving, critical thinking, and project management, which are essential in many industries
- To improve communication skills by presenting project findings and results to peers, faculty, and industry professionals.
- To showcase creativity and innovation by developing a unique and original project idea
- To be able to work better in a team

1.6.3 Technologies

The tools and Technologies we will be using are:

- Unity Engine
- Visual Studio

- Photon Engine
- Adobe Photoshop

1.6.4 Limitations

A few limitations of our project are:

- Lack of Depth: Due to the nature of endless runner games, they often lack the depth and complexity of other genres, such as RPGs or adventure games
- Limited Scope: Endless runner games are often limited in terms of their scope and gameplay mechanics, which can limit their appeal to certain players
- Technical Limitations: Developing an endless runner game with high-quality graphics and responsive controls can be challenging, particularly on mobile devices with limited processing power and memory

1.7 Feasibility Study

1.7.1 Risks Involved

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

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

1.7.2 Resource Requirements

- Computer or Laptop
- Ram 12 GB or above
- IOS and Play store registration fee

1.7.3 Programming Language

All the programming of our game (front-end plus back-end) 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.

1.7.4 Multiplayer Programming

For multiplayer programming i.e., making our game multiplayer photon engine will be used with C# programming language.

1.8 Introduction to Unity Engine

Unity is a cross-platform game engine initially released by Unity Technologies, in 2005. The focus of Unity lies in the development of both 2D and 3D games and interactive content. Unity now supports over 20 different target platforms for deploying, while its most popular platforms are the PC, Android and iOS systems. In our project unity will be used in a number of ways such as

- Game Design: Unity's built-in game design tools, such as the visual editor, can be used 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 used to create characters, obstacles, power-ups, and other game elements
- Physics Engine: Unity's physics engine can be used to simulate realistic movements and interactions between game objects, such as gravity, collisions
- Testing: Unity provides various testing tools, such as the Unity Test Runner and Profiler, which can be used to test game mechanics, optimize performance, and ensure the game runs smoothly on different devices

Figure 1.1: Unity Engine Logo

Chapter 2

Existing Games

2.1 Endless Runner Game

Endless Runners feature a perpetually moving character that players should navigate around obstacles. These games might feature levels with a beginning and end, or they will never finish.

2.2 Subway Surfers

You are some graffiti "artist" who's running away from a security guard (and his dog). You jump over or dodge obstacles along the way and collect coins in the meantime. You have three lanes to run in and can swap between them by swiping across the screen. Some routes will have obstacles, while others will have coins. You can also climb up on a train's ramp to run along the trains' roofs. You can duck under some ramps by swiping down and jump over them by swiping up. When you hit a smaller obstacle, the guard gets a chance to catch up. If you hit a train, it's game over, and you have to start from scratch. After a while, the game ramps up, increasing the speed at which your game character runs. Your success in the game pretty much depends on how fast you can swipe. The coins you collect during each of your runs can be used to unlock more content, like gameplay improvements or cosmetics. They also present a scoring system by themselves, although there's another, separate scoring system based on how far you've managed to run.[1]

Figure 2.1: Subway Surfers logo

2.3 Temple Run: Classic

The player controls an explorer who has obtained an ancient relic and runs from demonic monkey-like creatures chasing him. While the character is running, the player can tilt their device left or right to move the character to either side of the screen to collect coins and/or avoid obstacles. There are three types of coins to be found while the character is running: gold, red, and blue. A gold coin will only add one coin to the player's total number of coins. Red coins are worth two coins, while blue coins are worth three. The coins can be used to buy and then upgrade power-ups and/or other characters. Coins can also be bought by the player through in-app-purchases with payments of actual money. If the player wishes to turn left or right, the touch screen can be swiped in the corresponding direction. If the player wishes to jump over an object, the screen can be swiped upwards and if they wish to slide under an object, the screen can be swiped downwards.[2]

Figure 2.2: temple run 1 logo

2.4 Temple Run 2

Temple Run 2 is the vastly improved graphics. Vibrant colors and much more detailed environments add to the appeal of the lost city in the sky setting. Temple Run 2 benefits from retaining the original game's control mechanics. Players must utilize a tried and tested swipe control system. Swiping left and right to make turns, users have to swipe up and down to avoid various hazards along the route, including fire-breathing statues and precariously placed rocky outcrops. Tilting the device controls, the mine carts as well as letting users pick up power-ups and the coins and gems that make up the currency for Temple Run 2.[3]

Figure 2.3: temple run 2 logo

2.5 Sonic Dash

Dash offers up a Sonic-themed take on hit endless runner Subway Surfers. Sonic runs through the environment automatically and you swipe left or right to move between three distinct lanes. Swiping down activates an enemy-destroying spin dash and swiping up jumps. You'll dash below overhangs and through enemies, jump over spikes and wind your way through the environment until eventually the action gets too fast and you make a mistake.[4]

Figure 2.4: Sonic Dash logo

2.6 Comparison Table

Serial Number	**Comparison Points**	**Temple run**	**Subway Surf**	**Sonic Dash**
1	Year of release	August 4, 2011	May 23, 2012	March 7, 2013
2	Creator	Imangi Studio	Kilo and SYBO games	Hardlight and Sega
3	Engine Used	Temple run engine iOS Unity (Android)	Unity Engine	Unity Engine
4	Status	Active	Active	Active
5	Features	Classic 3D endless runner game	3D endless runner game with improved graphics	Developed after a popular game Sonic the Hedgehog
6	Problems	Single playing mode, always running in the same environment	Single playing mode	Slow gameplay, always running in the same environment

Table 2.1: Table of Comparisons

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 in order to proceed further.

3.4 Design

In 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, power ups etc.

3.5 Implementation

In the implementation or development phase the actual coding of all the levels will be conducted on the basis of 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 especially a multiplayer game. 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 images adobe photo shop was used. We learnt basic photo shop for this purpose
- The main editor the tools we used was 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 such as after starting the game which mode the user decided to play in that is single player or multiplayer. Which character the user decided to play with and only the characters that the user has purchased will be available to the user while others will remain locked. After the game ends all the coins are added to his purse so that he can purchase other characters.

3.9 Functional Requirements

Functional requirements are a type of requirement that describe what a system, product or service must be able to do or perform to satisfy the needs of its users or stakeholders. Some functional requirements of this project are listed below

1. **Single Player**
   Input: Select single player mode
   Output: User will be able to play the game in single player
   Process: Verification of the choice of game mode will be made
   Description: It will be verified that what game mode the user has selected and the game will start according to the user's choice

2. **Multiplayer**
   Input: Select multiplayer mode
   Output: User will be able to play in multiplayer mode
   Process: Verification of the choice of game mode will be made
   Description: It will be verified that what game mode the user has selected and the game will start according to the user's choice

3. **Character change**
   Input: Select character change option
   Output: User will be able to change his character
   Process: Verification of the selected option will be made
   Description: After the user selects the character change option, they will be able to select from a list of characters and play with a character of their choice

4. **Game Shop**
   Input: Select shop option in the game
   Output: User will be directed to the in-game shop
   Process: Verification of the selected option will be made
   Description: After the user selects the shop, they will be able to purchase different items from the shop

5. **Pause menu**
   Input: Click on pause option
   Output: Game will be paused
   Process: User can pause the game in single player mode
   Description: If the user wants to pause the game for any reason, they can do so in single player mode and resume the game afterwards from where the user left off

6. **Scenery change**
   Input: Click on scenery change option

Output: 3 options will appear: day, night, and evening
Process: User can choose a scenery from these three modes
Description: If the user wants to change the scenery of the game, they can do so by simply going to the scenery change option and choosing a scenery of their 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:

- Performance: This refers to the ability of the system to handle a certain volume of transactions or requests, and to respond quickly to user inputs.
- Usability: This refers to the ease of use and accessibility of the system, as well as its overall user experience.
- Reliability: This refers to the ability of the system to function properly and consistently over time, without experiencing downtime or errors.
- Security: This refers to the measures in place to protect the system from unauthorized access or other security breaches.
- Scalability: This refers to the ability of the system to handle an increasing workload or user base, without a significant loss in performance.
- Maintainability: This refers to the ease of making changes or updates to the system, as well as the ease of troubleshooting and repairing issues that arise.

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 case. It is a methodology used in system analysis to identify, clarify and organize system requirements. Also fins 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 Use case description for Race 'em Multiplayer game

Following are the use case diagrams for different functional requirements of the game:

3.12.1 Single Player

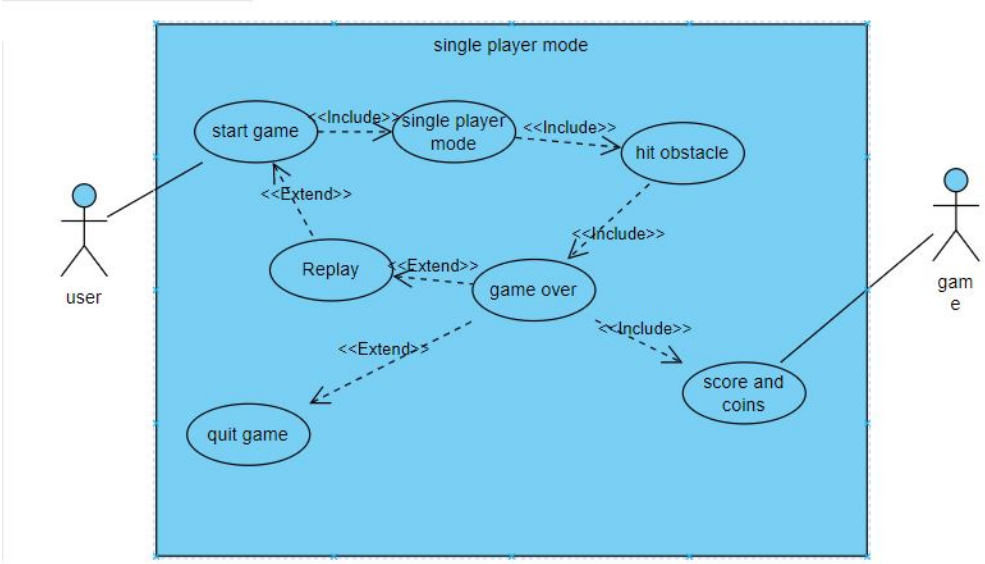

Figure 3.1: Single player use-case

3.12.2 Multiplayer

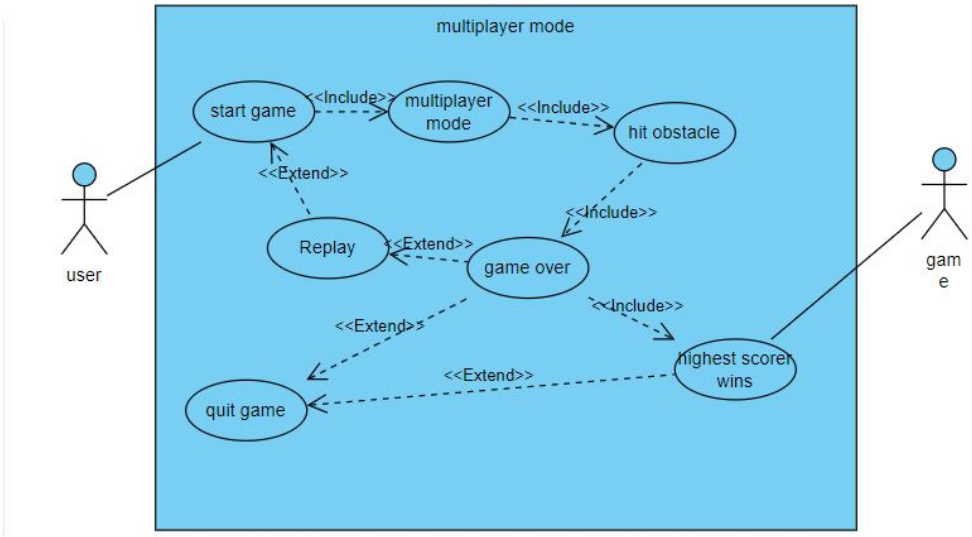

Figure 3.2: Multiplayer use-case

3.12.3 Character Change

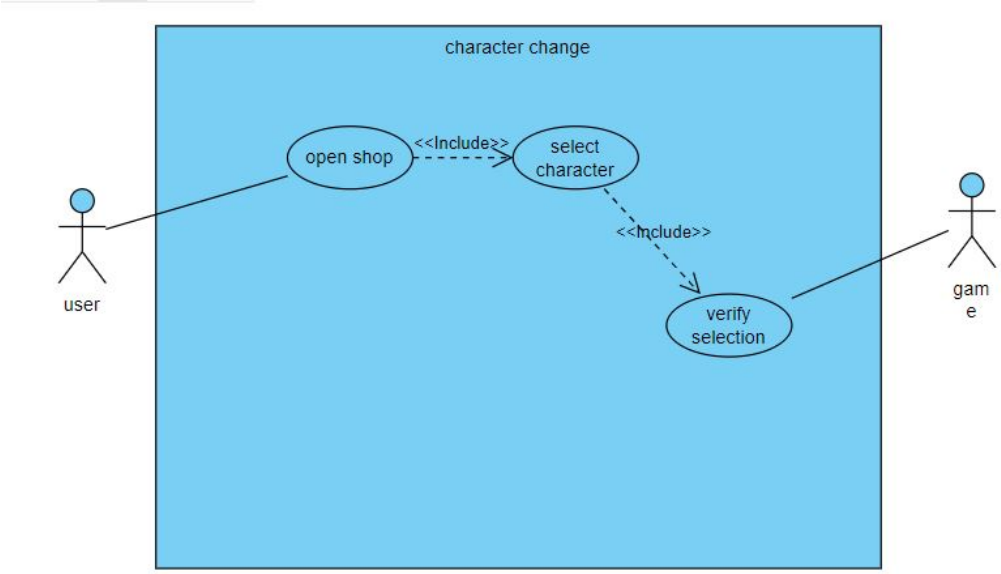

Figure 3.3: character change use-case

3.12.4 In-Game shop

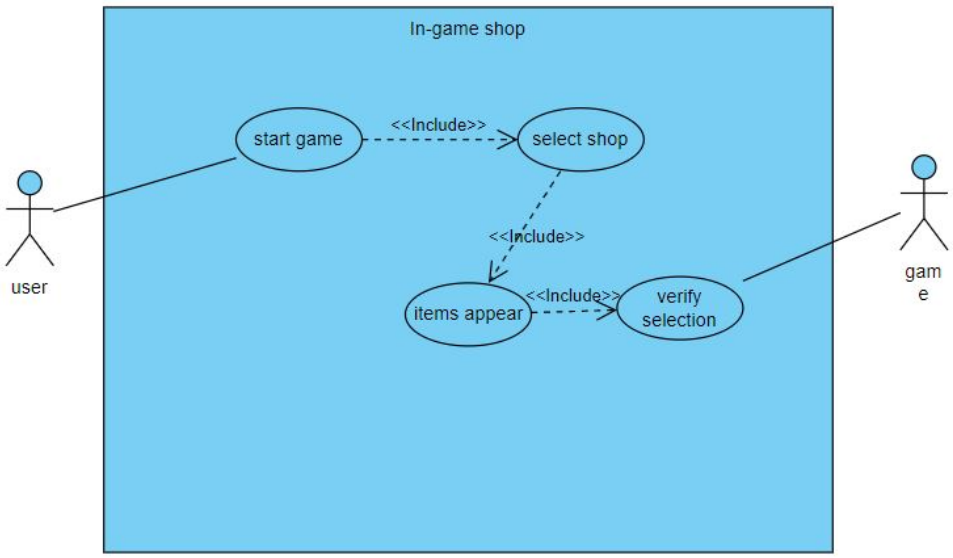

Figure 3.4: in-game shop use-case

3.12.5 Pause Menu

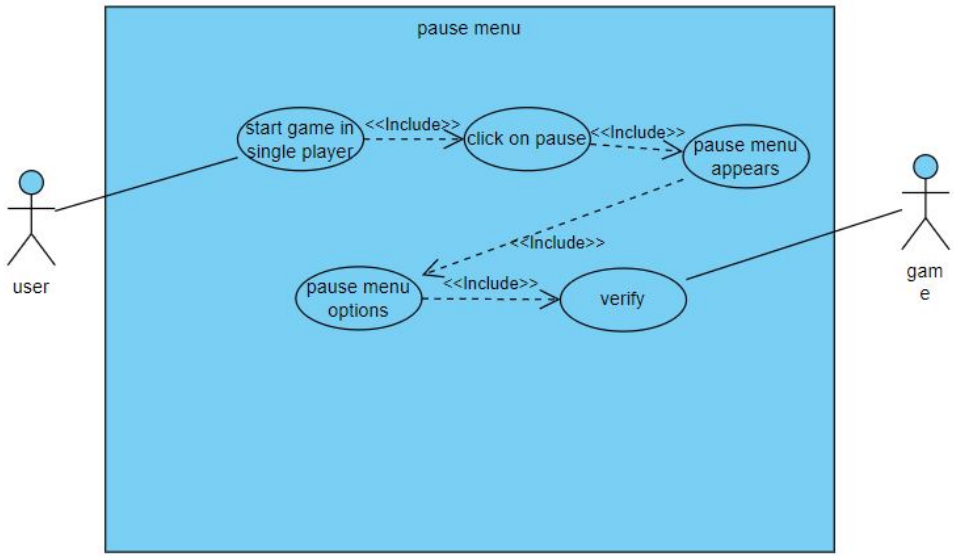

Figure 3.5: pause menu use-case

3.12.6 Scenery Change

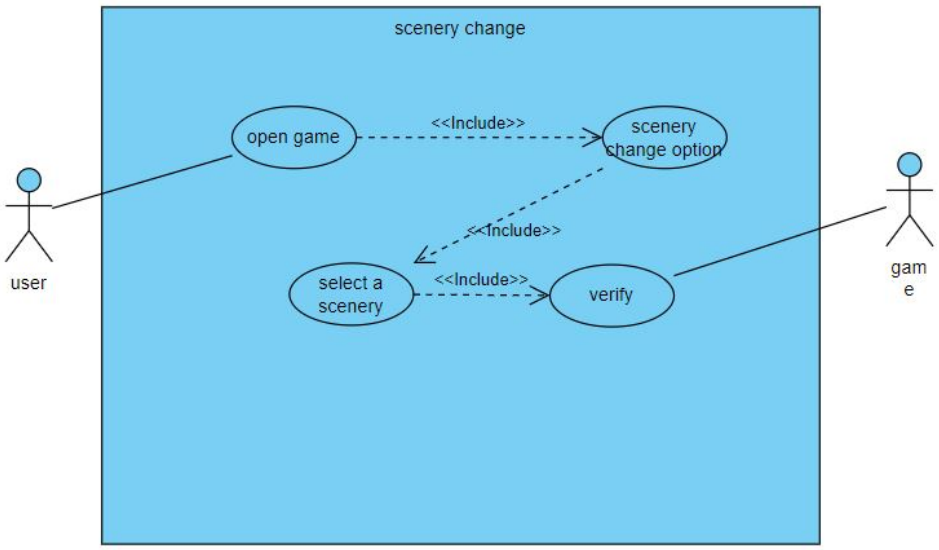

Figure 3.6: Scenery change use-case

3.13 Flow Chart

The following flowchart shows the flow of control in our project in both single player as well as multiplayer mode.

3.13.1 Single Player Mode

Start
The game will begin
Character starts running
The character will run while players dodge different obstacles
Could not Dodge
If the player is unable to dodge and trips the game is over
Do you wish to continue?
This will confirm if the user wants to play again

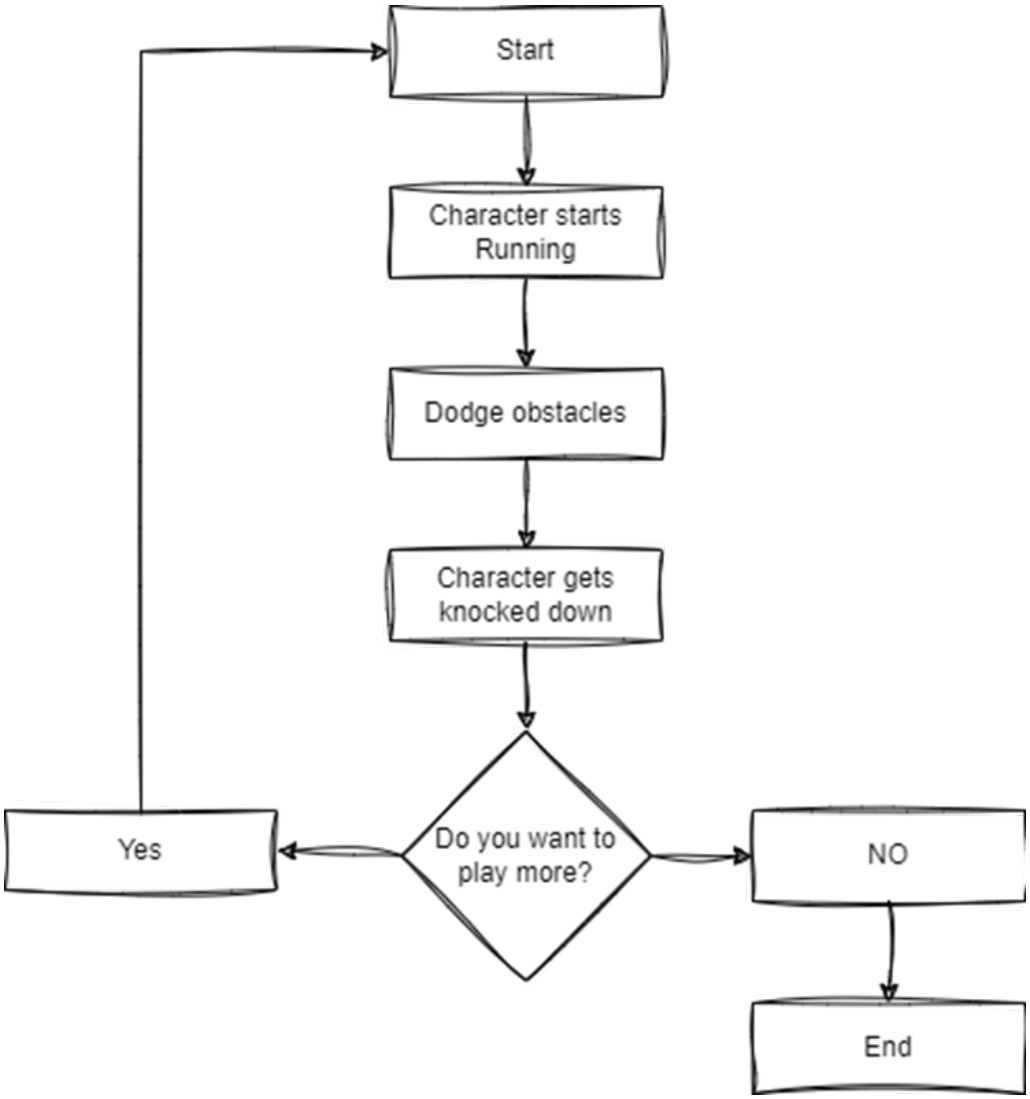

Figure 3.7: Single player flowchart

3.13.2 Multiplayer Mode

Start
The game will begin
Character starts running
The character will run while players dodge different obstacles
Could not Dodge
If the player is unable to dodge and trips the game is over
Winner
The player with the highest score wins
Do you wish to continue?
This will confirm if the user wants to play again

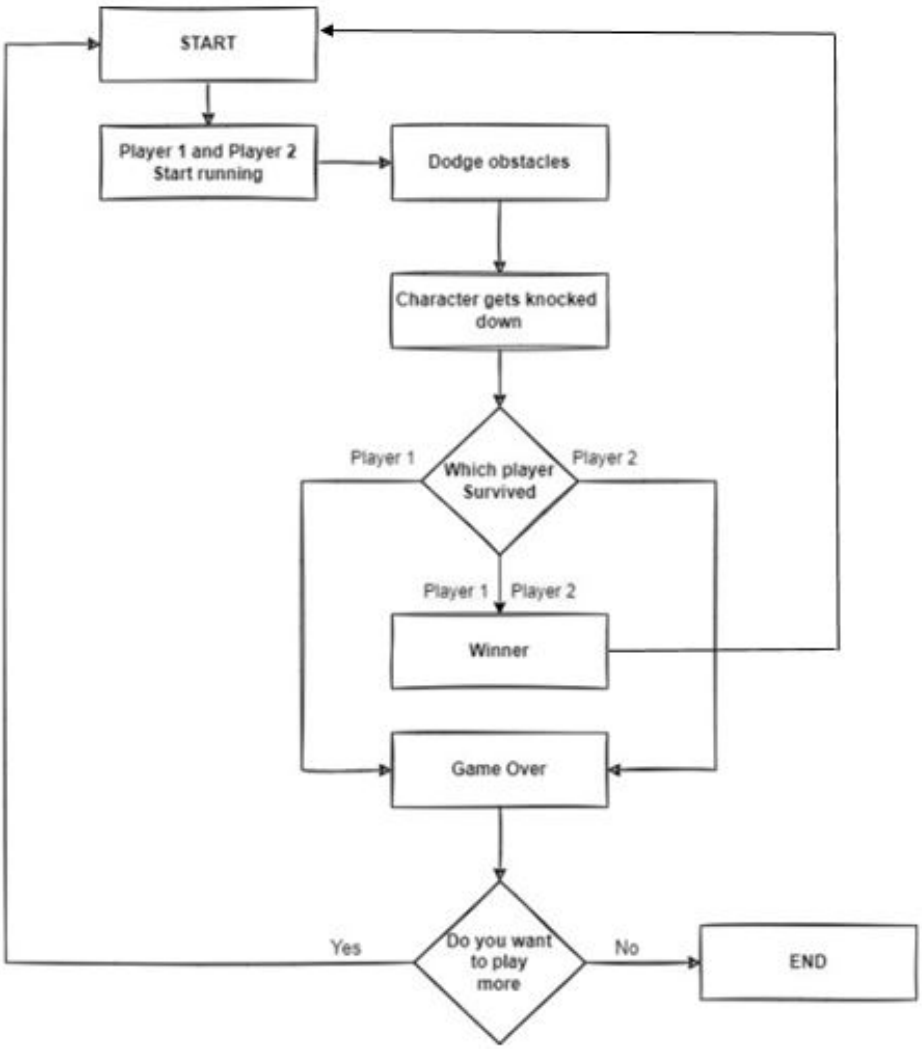

Figure 3.8: Multiplayer flowchart

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 interactive and simple. It consists of user friendly interface which contains the main menu for interaction. There is another menu for option in which instructions of the game is written, it will help the player to play game. There is an Exit menu to quit the game. The game is third person endless running, in which the character has to run dodge obstacles and make a score. Hitting any obstacle will cause the game to end.

4.1.1 Single Player Architecture Diagram

Following is the architecture diagram of our proposed system in single player mode

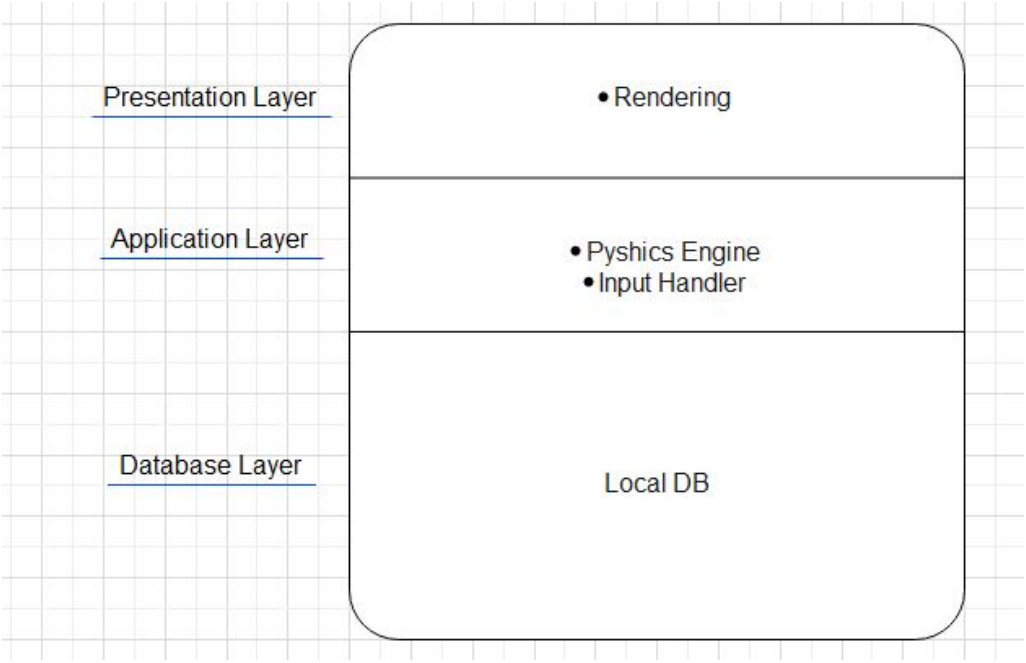

Figure 4.1: Single player architecture diagram

Explanation:

- **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

- **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.

4.1.2 Multiplayer Architecture Diagram

Following is the architecture diagram of our proposed system in multiplayer mode

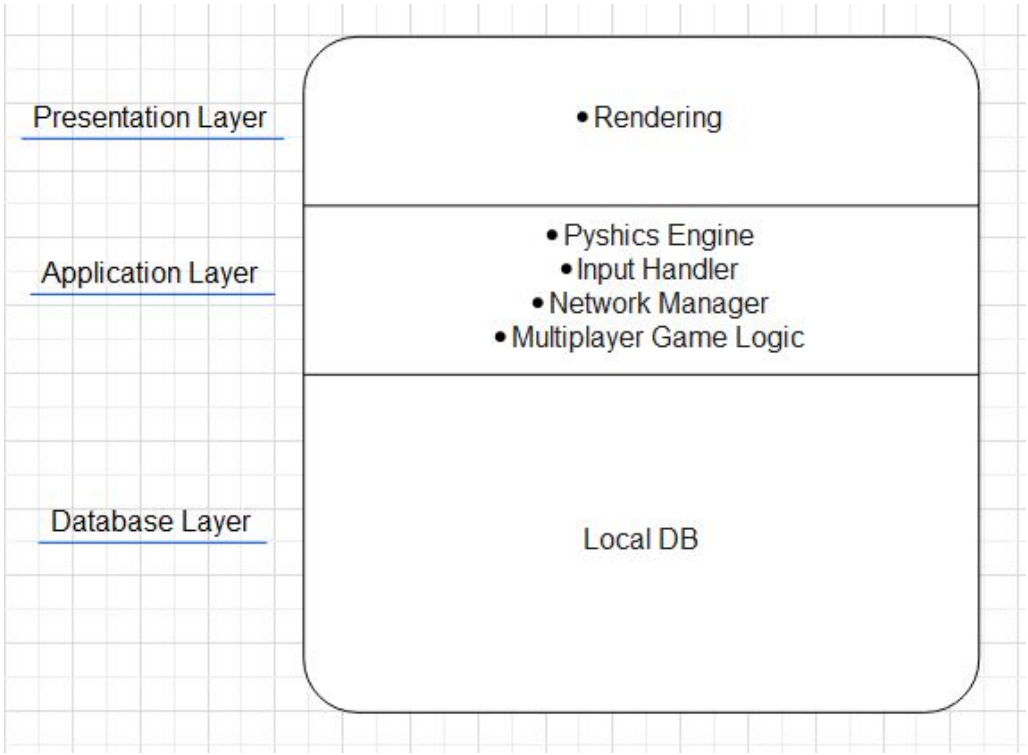

Figure 4.2: Multiplayer architecture diagram

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

- **Network Manager:** 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

- **Multiplayer Game Logic:** This subsystem handles the specific game logic for multiplayer mode, including player synchronization, scoring, and other game rules that are specific to multiplayer mode.

4.2 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 has to dodge obstacles and make a higher score as compared to the other player. Hitting any obstacle will cause the character to stumble and game to end.

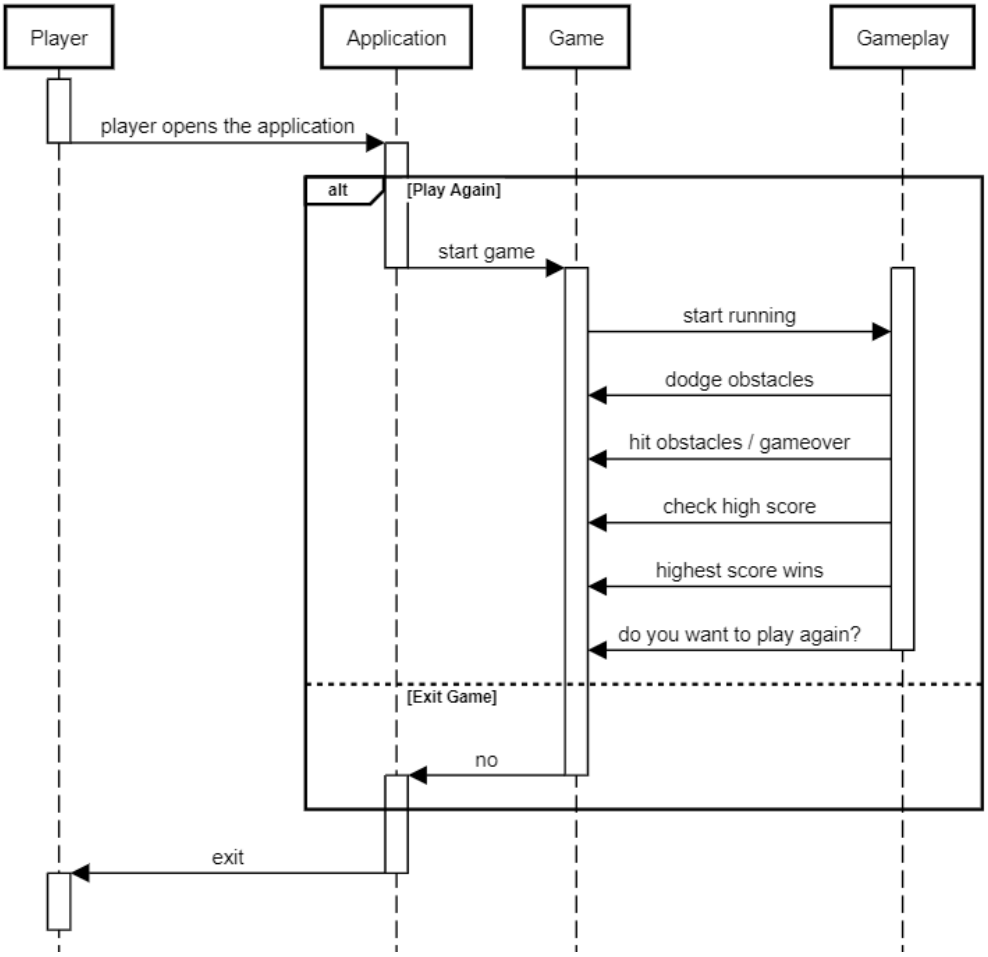

Figure 4.3: Sequence diagram

4.3 Character Animations

By using animations, we are going to move the still characters to make them do what is required of them. There are a few basic animations in our game that are the basic movements of the character to dodge the obstacles in his path. These animations are:

4.3.1 Running

This is the most common animation that will be used most commonly in the game. In this animation the character will start to run.

Figure 4.4: Running Animation

4.3.2 Dying

This animation will occur when the character hits an obstacle and the game is over.

Figure 4.5: Dying Animation 1

Figure 4.6: Dying Animation 2

4.3.3 Jump

This animation will show the character jumping in the air. This will be used to dodge obstacles that require jumping to dodge them.

Figure 4.7: Jump Animation 1

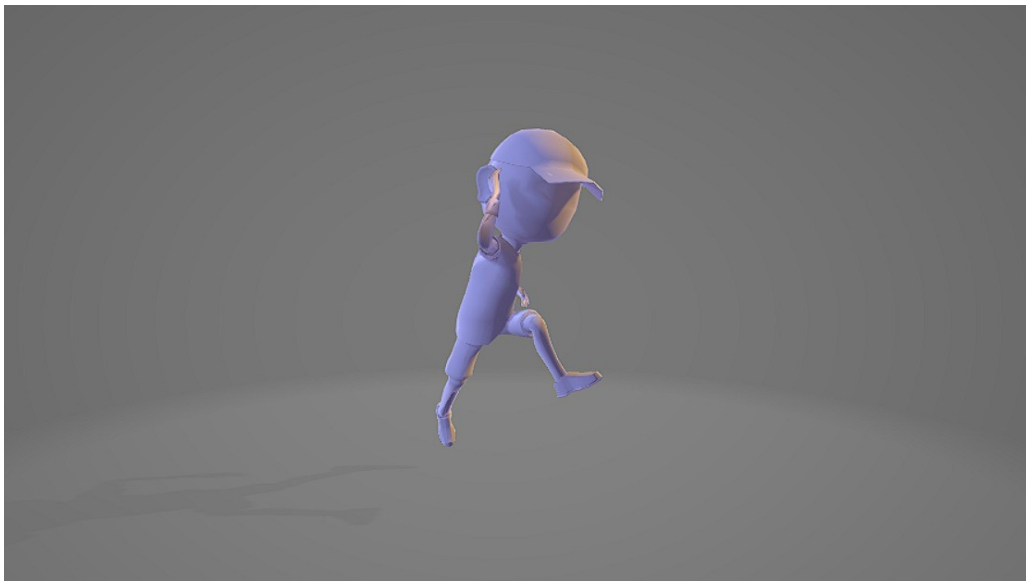

Figure 4.8: Jump Animation 2

4.3.4 Roll to Run

In this animation the character will roll either to dodge obstacles or when the user wants. Roll and slide animation will be used at random by the game.

Figure 4.9: Rolling Animation 1

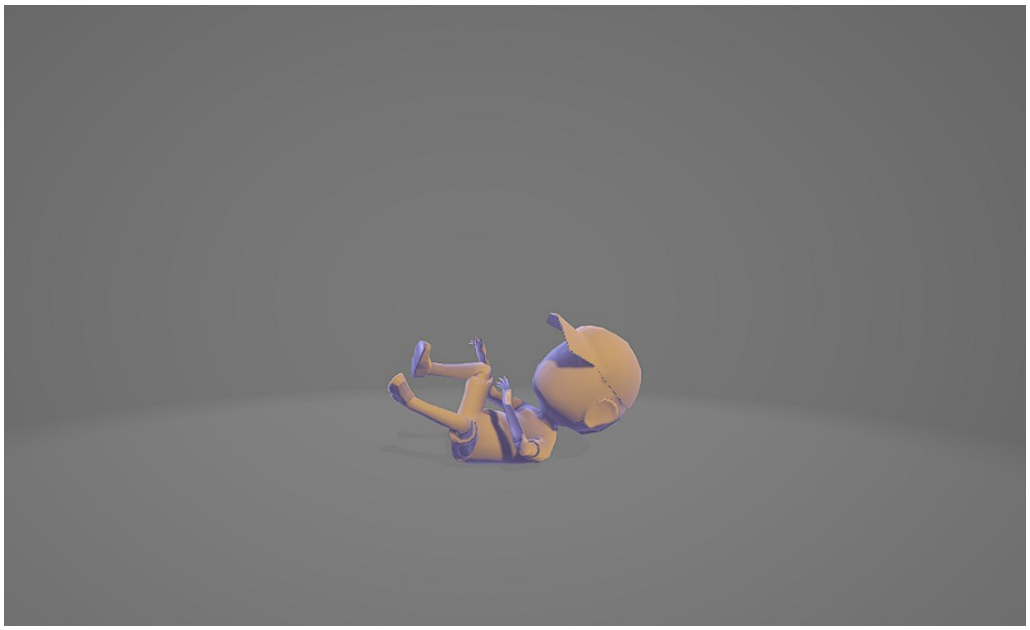

Figure 4.10: Rolling Animation 2

Figure 4.11: Rolling Animation 3

4.3.5 Slide

Slide animation will get the character to slide used at random with roll animation

Figure 4.12: Sliding Animation 1

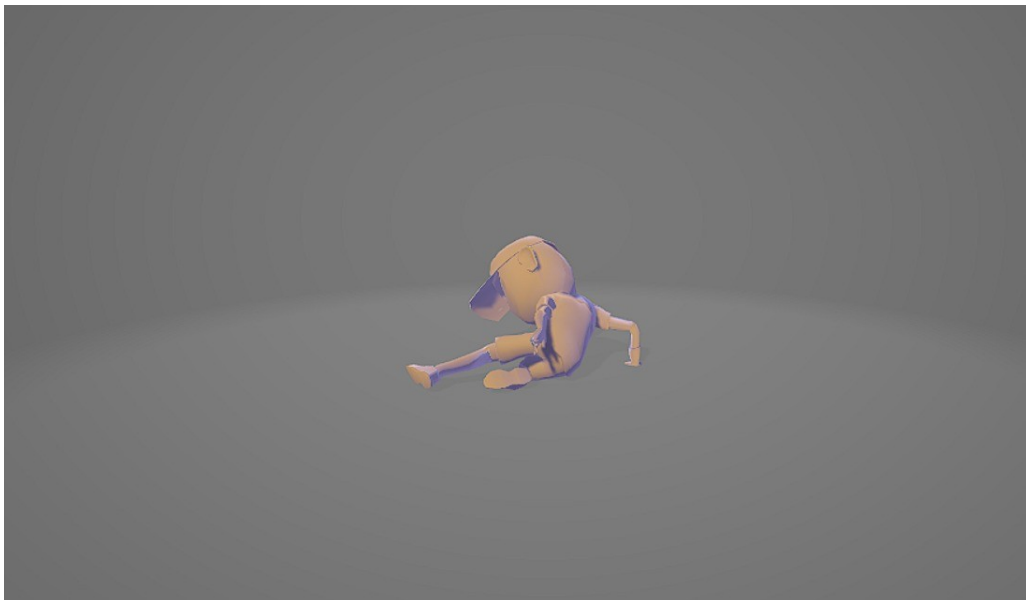

Figure 4.13: Sliding Animation 2

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
- Photon Engine
- 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 noncoding part as our front-end design.

5.4 Characters in mixamo

We start our journey towards making our game from here. We firstly made a rough design of how we wanted our characters to look like and then after that we used mixamo to choose characters and import them in our game.

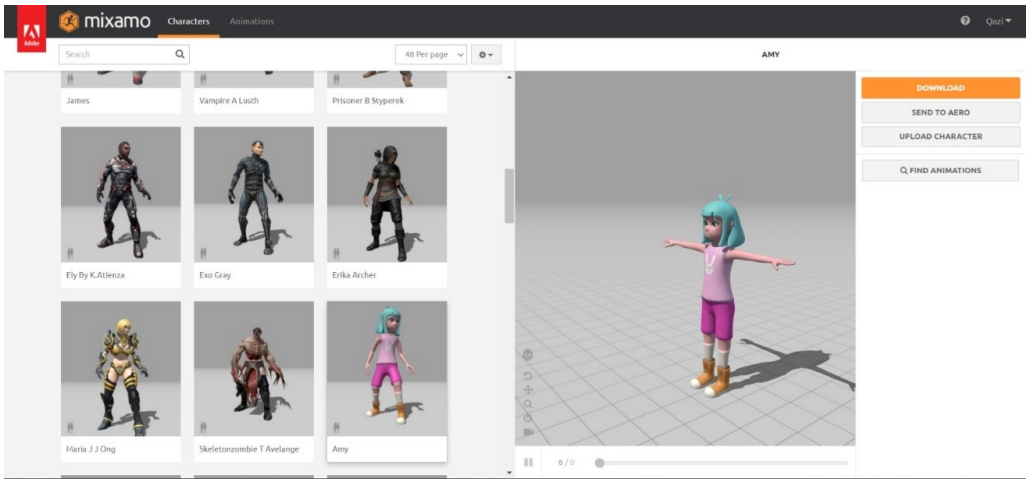

Figure 5.1: Character in mixamo

5.5 Photon Engine

Photon Engine is a multiplayer game development framework and backend service. It provides tools for creating real-time multiplayer games, handles communication between players, and offers features like networking, matchmaking, and platform support. It simplifies the development of multiplayer games and allows for smooth, scalable, and engaging experiences.
We also used photon engine for our multiplayer game development.

Figure 5.2: Photon Engine Logo

Our game in photon:

The following shows our game's multiplayer creation in photon engine.

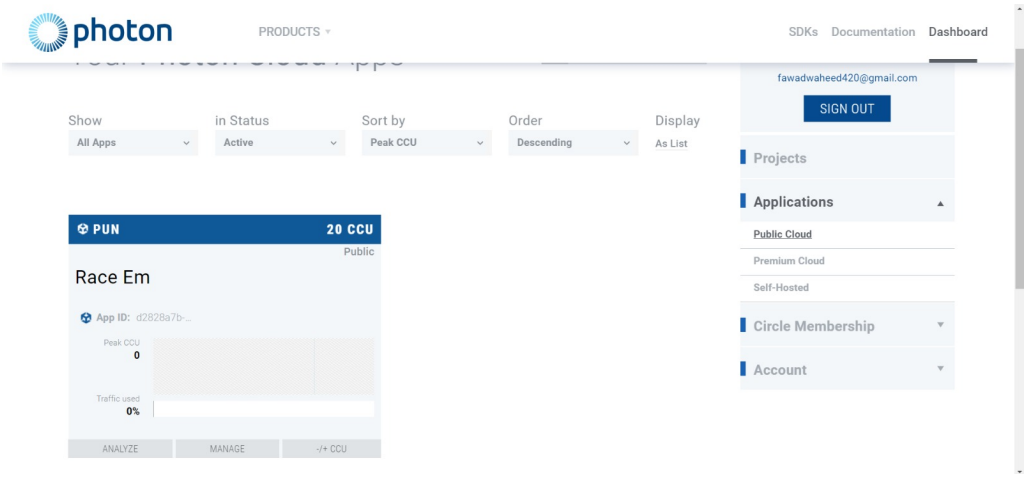

Figure 5.3: Our game in photon

5.6 Game Environment in Unity

Since we made our environment based on road and added road side obstacles, first we imported the fbx 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 whole terrain. The figure shows the game 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 5.4: Game Environment 1

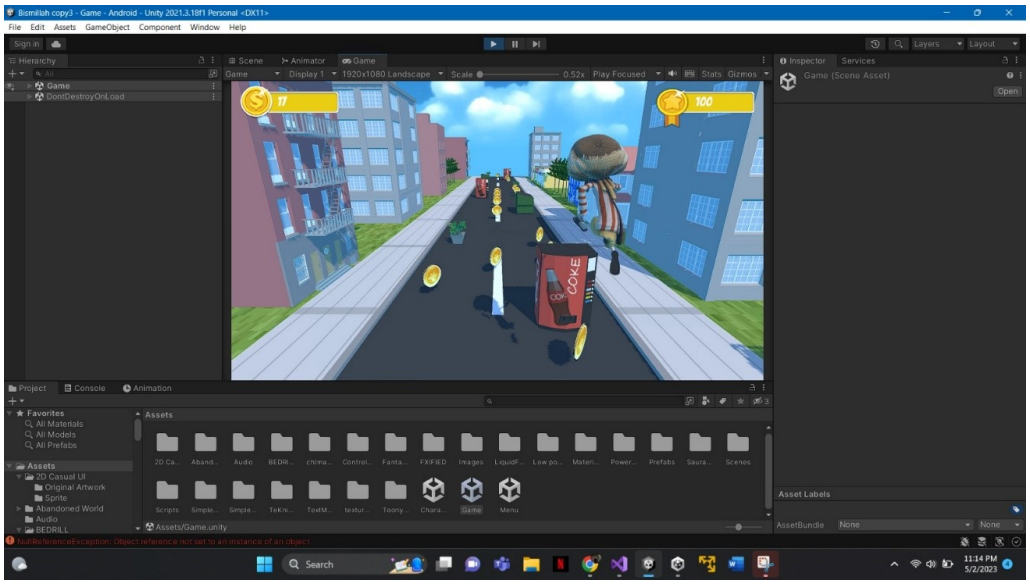

Figure 5.5: Game Environment 2

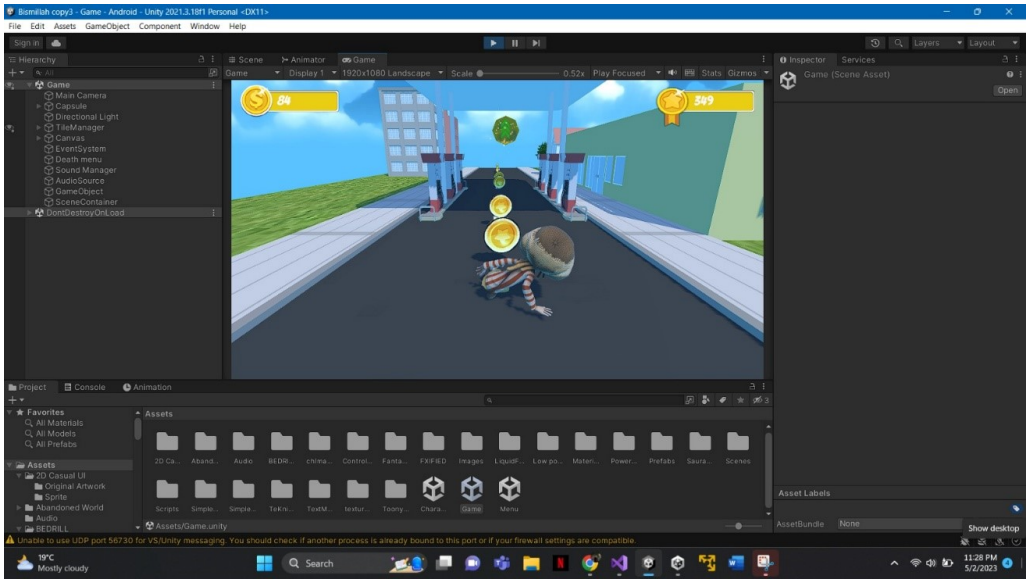

Figure 5.6: Game Environment 3

5.7 Other Options

Following are some of the other modules of our project. These modules provide the user with more options regarding their gameplay and experience.

5.7.1 Start Menu

This is the menu that will be displayed once the game is turned on. It provides different options such as in the top right corner your high score and total coins will be displayed on the bottom left in-game shop icon is available and in the middle of the screen the start button is available.

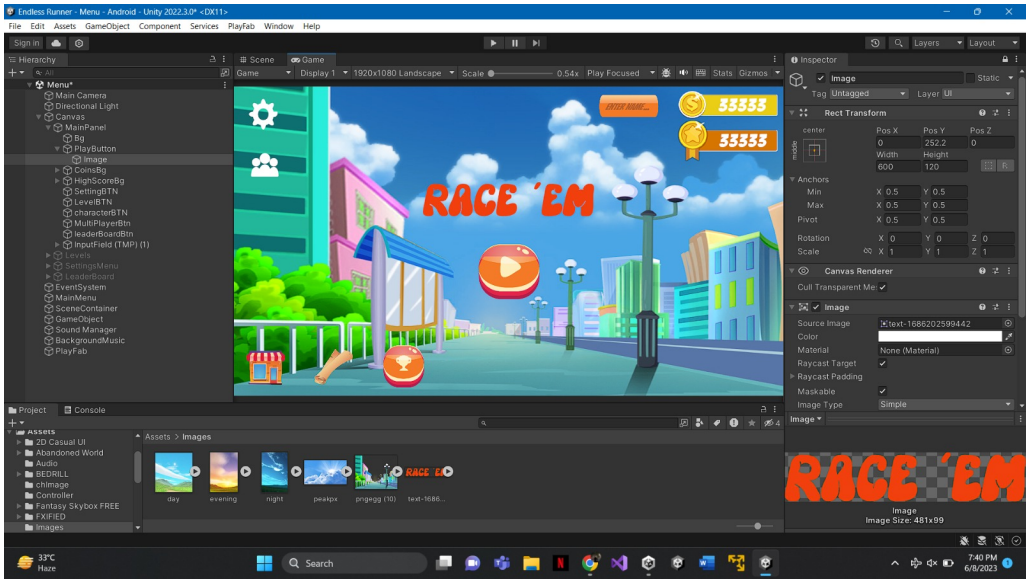

Figure 5.7: Start Menu

5.7.2 In-game shop

This is the icon on the bottom left side of the start menu. In the shop user will be able to purchase new characters with the coins that they collect from playing the game. You can also change your character from the shop (note that you will be able to select only the characters that you have purchased if you want more characters you will first have to purchase them from the shop using coins)

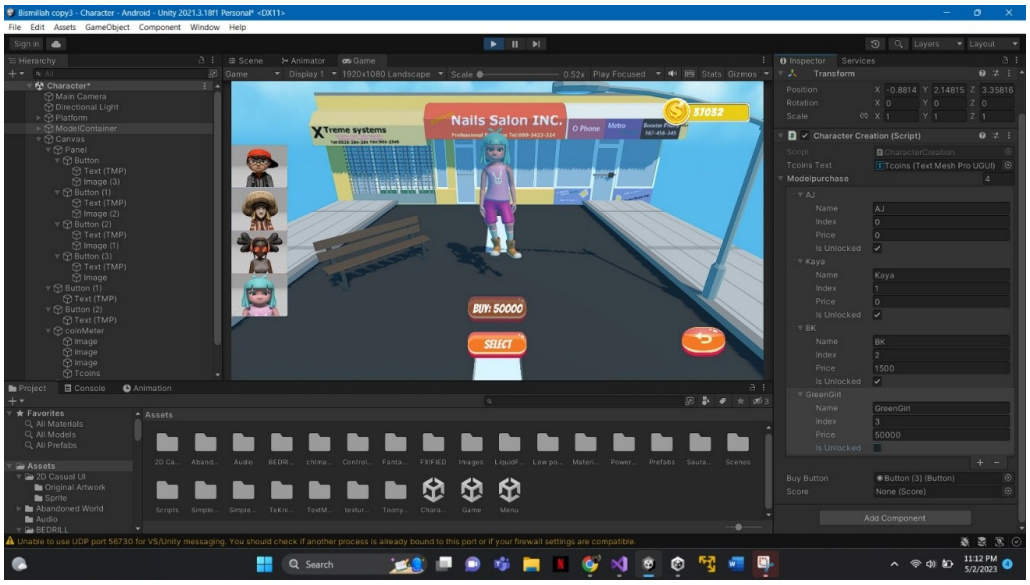

Figure 5.8: Unpurchased Character

The above figure shows the option of purchasing a character.

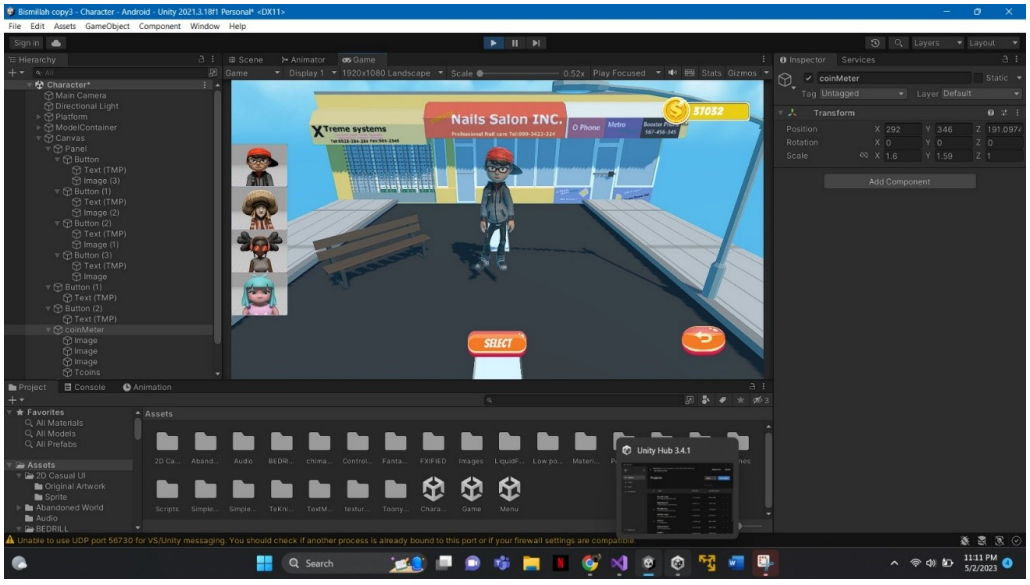

Figure 5.9: Character Purchased

The above figure shows the option to select the character you have already purchased.

5.7.3 Pause Menu

This is the menu that will be displayed if you choose to pause the game. It will provide you with three options resume the game, quit the game and go to main menu. (note that game will only be paused in the single player mode)

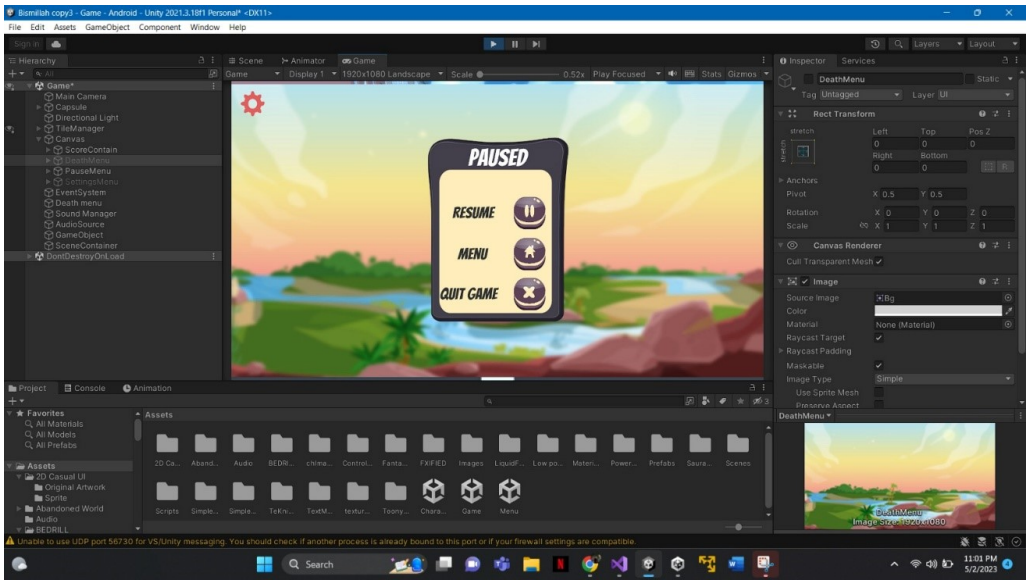

Figure 5.10: Pause Menu

5.7.4 End Menu

This is the menu that will be displayed once your character is eliminated. This menu shows your score, the coins you collected and the options of playing again or going to the main menu.

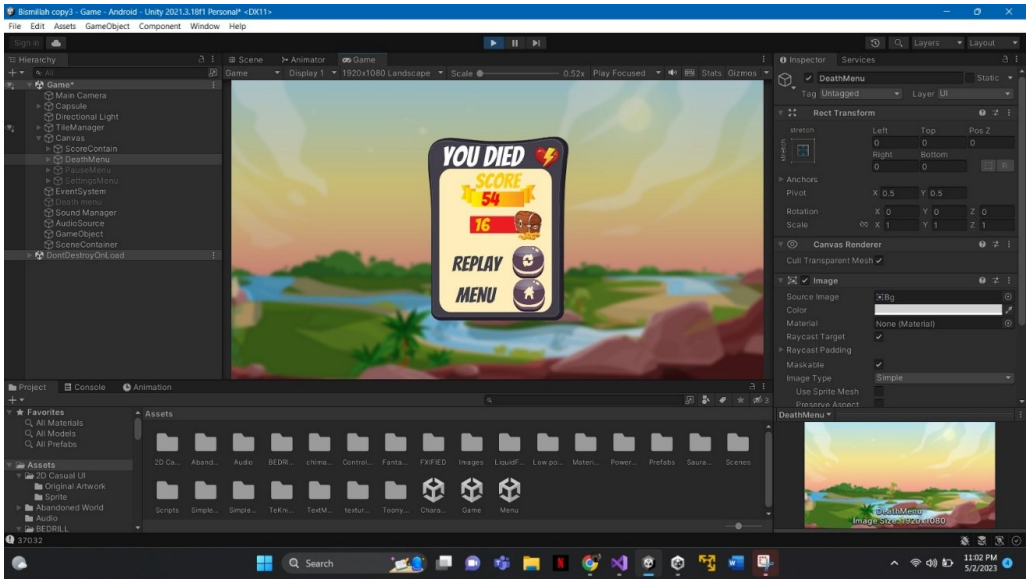

Figure 5.11: Death Menu

5.7.5 Scenery Change

From this option user will be able to change the scenery of the game. The user will be able to select day, night or evening scenery based on their preference.

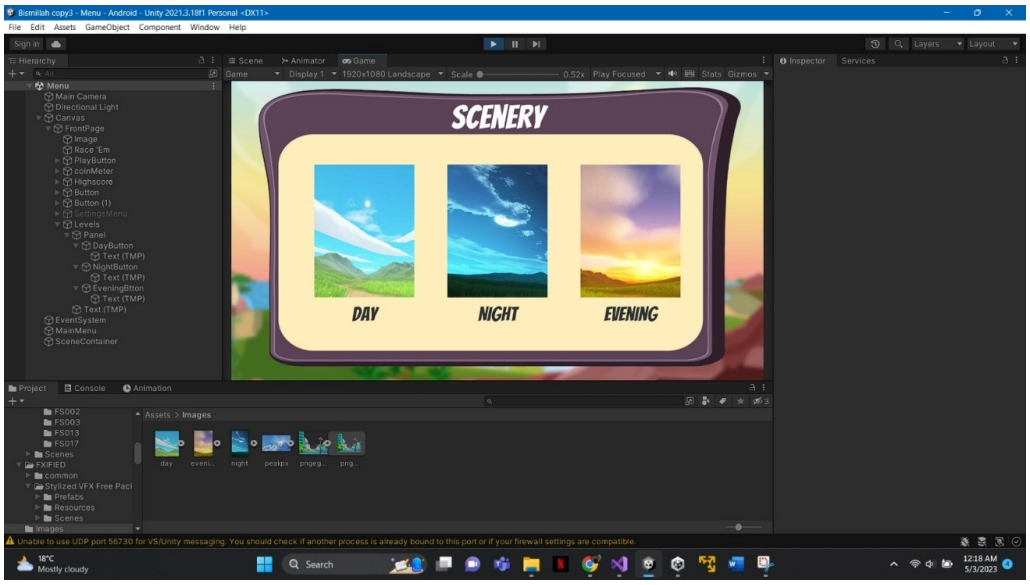

Figure 5.12: Scenery Change Option

Day Scenery

Figure 5.13: day scenery

Evening Scenery

Figure 5.14: Evening scenery

Night Scenery

Figure 5.15: night scenery

5.7.6 Multiplayer

Through this option user will be able to play with their friends in multiplayer mode. Firstly user will select the multiplayer option by clicking on the multiplayer button in the main menu. Then one player will create a server and the other will join by typing in the name of the server(case sensitive). After both players have joined the game will start and which ever player scores the more wins.

opening multiplayer

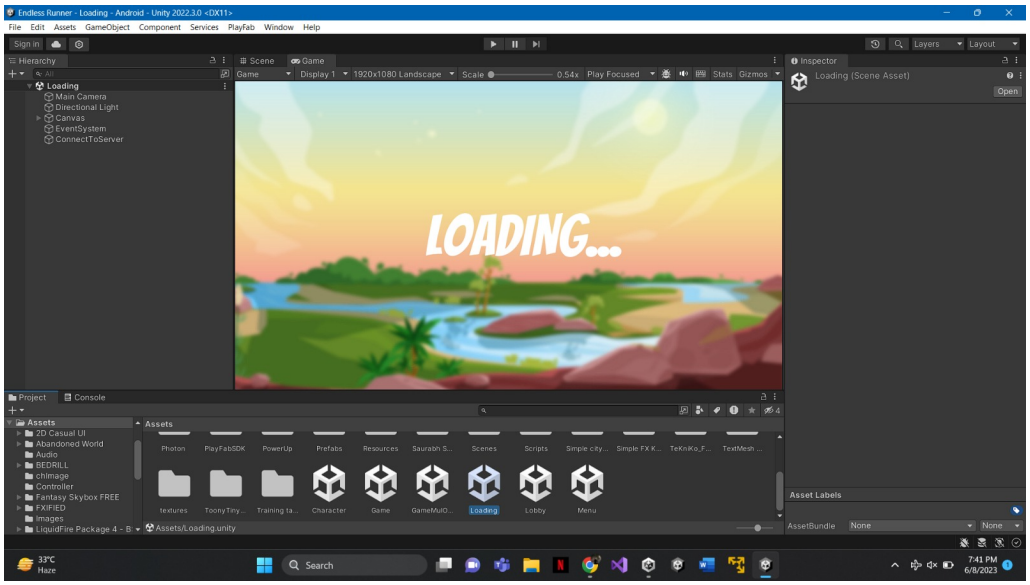

Figure 5.16: Opening multiplayer mode

Multiplayer mode

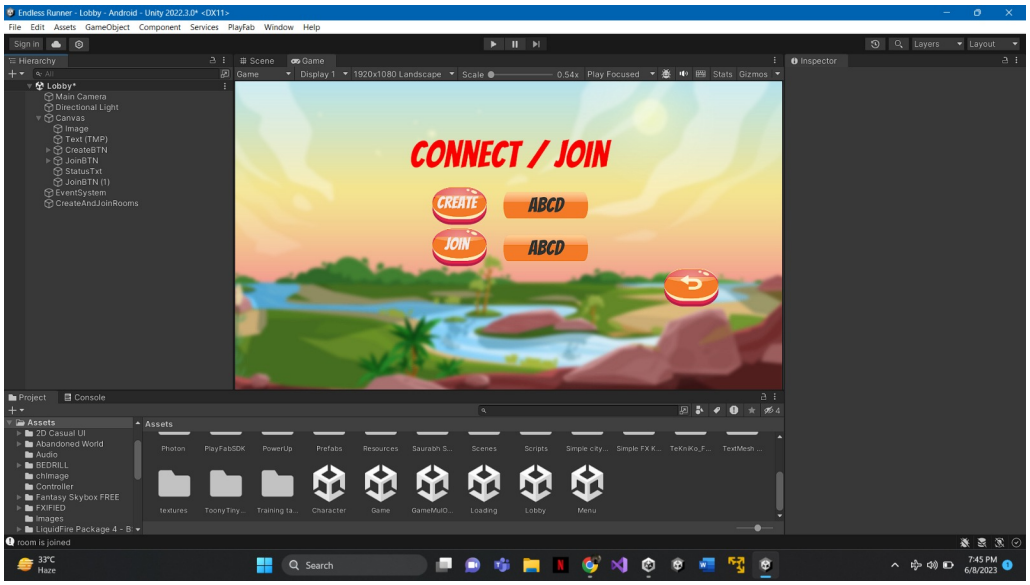

Figure 5.17: Multiplayer mode

Creating Multiplayer Server

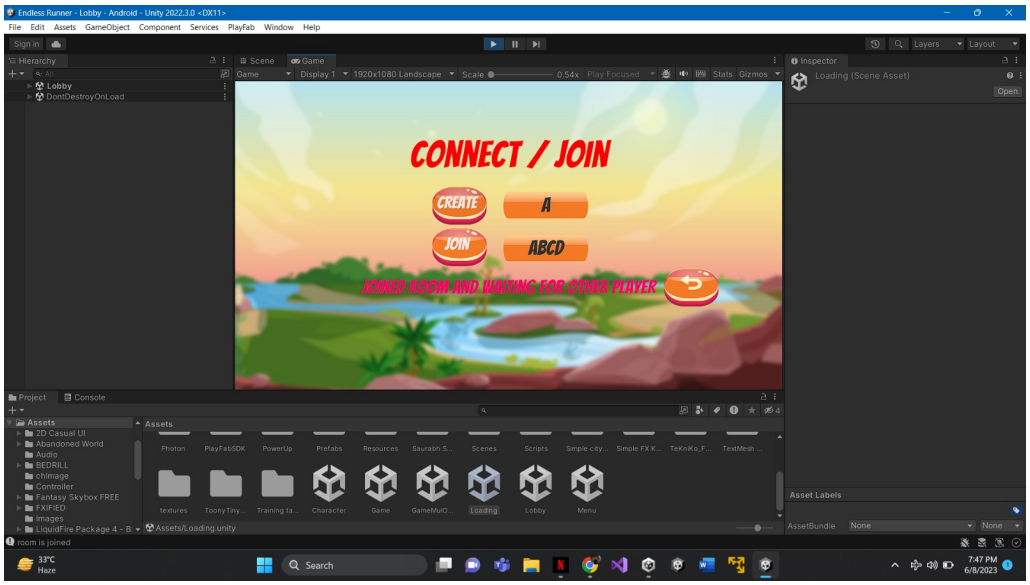

Figure 5.18: Creating server

Multiplayer Gameplay

During the multiplayer run your score and the other player's score will be displayed on the top right corner of the screen.

Figure 5.19: Multiplayer game play

Multiplayer Death Screen

After your run ends in the death menu the scores of the players will be compared.

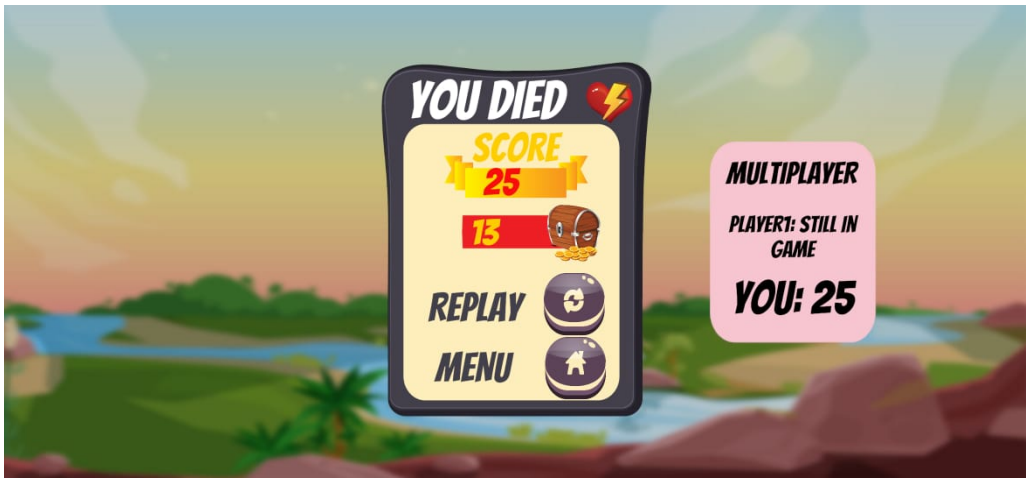

Figure 5.20: Multiplayer Death Screen

5.7.7 Leader Board

In this feature you will be able to see who has the highest score and compete against them to see who can score the highest.

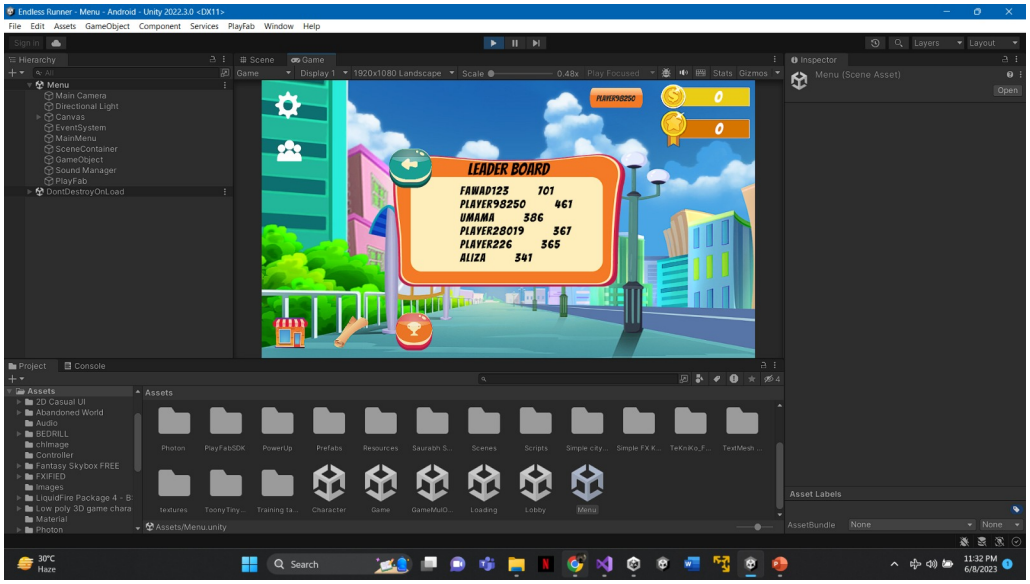

Figure 5.21: Leader Board

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 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

1. **Manual Testing**: This involves testing the game manually by playing it and observing its behavior. It is a time-consuming but the most successful process that requires testers to play the game multiple times to identify bugs and glitches. By playing the game themselves the testers can better understand the game from a user's point-of-view. 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.

2. **Automated Testing**: This involves the use of automated testing tools to simulate user interactions and verify the functionality of the game. Automated testing can be used to test the game across multiple devices and operating systems, which can save time. For this purpose, the built-in unity tester called unity test runner was used.

3. **Functional Testing**: This involves testing the game's functionality, such as the movement of the character, collision detection, and scoring system. For this we made different test cases to keep track of all the progress that we made as well as to track if our project is working properly.

6.2.1 Test Cases

6.2.2 Singleplayer Mode

Test Case ID	**SPM-001**
Test Case Description	Test the single player mode selection
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed
Test Data	1. User id 2. User Password
Test Scenario	Verify by selecting single player mode the game should begin in the selected mode
Step Details	1. Open the game 2. Select single player mode

Expected Results	1. Game should open 2. Game should start in the selected mode
Actual Results	1. Game opened 2. Game started in single player mode
Status	1. Pass 2. Pass

Table 6.1: Singleplayer Test Case

6.2.2.1 Singleplayer Mode Test Case To Fail

Test Case ID	**SPM-002**
Test Case Description	Test that single player mode selection is working properly or not
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed

Test Data	1. User id 2. User name
Test Scenario	Verify that if we select some other mode does the game still start in single player or not
Step Details	1. Open the game 2. Select multiplayer mode instead of single player
Expected Results	1. Game should open 2. Game should not begin in single player mode
Actual Results	1. Game opened 2. Game did not start in single player mode
Status	1. Pass 2. Pass

Table 6.2: Singleplayer mode Test Case to fail

6.2.2.2 Multiplayer Mode

Test Case ID	**MPM-001**

Test Case Description	Test that multiplayer mode selection is working properly or not
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Internet Access
Test Data	1. User id 2. User Password
Test Scenario	Verify that if we select multiplayer mode the game starts in the intended mode
Step Details	1. Open the game 2. Select multiplayer mode
Expected Results	1. Game should open 2. Game should begin in multiplayer mode
Actual Results	1. Game opened 2. Game started in multiplayer mode

Status	1. Pass 2. Pass

Table 6.3: Multiplayer Test Case

6.2.2.3 Multiplayer Mode test case to fail

Test Case ID	**MPM-002**
Test Case Description	Test that multiplayer mode selection is working properly or not
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Internet Access
Test Data	1. User id 2. User name
Test Scenario	Verify that if we select some other mode does the game still start in multiplayer mode or not

Step Details	1. Open the game 2. Select single player mode instead of multiplayer
Expected Results	1. Game should open 2. Game should not begin in multiplayer mode
Actual Results	1. Game opened 2. Game did not start in multiplayer mode
Status	1. Pass 2. Pass

Table 6.4: multiplayer mode Test Case to fail

6.2.2.4 Character Change

Test Case ID	**CC-001**
Test Case Description	Test the character change functionality
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023

Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. 2 or more characters should be purchased
Test Data	1. User id 2. Character Access 3. In-game shop access
Test Scenario	Verify that if we select another character does the change apply or not
Step Details	1. Open the in-game shop 2. Select a character that you have purchased
Expected Results	1. shop should open 2. Character should be selected and changes applied during run
Actual Results	1. In-game shop opened 2. The changes were applied successfully

Status	1. Pass 2. Pass

Table 6.5: Character change Test Case

6.2.2.5 Character change test case to fail

Test Case ID	**CC-002**
Test Case Description	Test the character change functionality is working properly or not
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed
Test Data	1. User id 2. In-game shop access
Test Scenario	Verify that if we select a character that we have not purchased the changes do not apply
Step Details	1. Open the in-game shop 2. Select a character that we have not yet purchased

Expected Results	1. shop should open 2. Game should not allow you to use that character
Actual Results	1. In-game shop opened 2. We are not able to select the character
Status	1. Pass 2. Pass

Table 6.6: Character change Test Case to fail

6.2.2.6 In-game shop pass

Test Case ID	**IGS-001**
Test Case Description	Test the in-game shop
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed

Test Data	1. User id 2. In-game shop access
Test Scenario	Verify that if we open the in-game shop does it open and we can purchase character successfully or not
Step Details	1. Open the game 2. Select the shop option from main menu 3. Purchase a character you have enough coins for
Expected Results	1. shop should open 2. Game should not allow you to use that character
Actual Results	1. game should open 2. shop should open 3. Character should be purchased
Status	1. Pass 2. Pass

Table 6.7: In-game shop pass test case

6.2.2.7 In-game shop fail

Test Case ID	**IGS-002**
Test Case Description	Test the in-game shop
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed
Test Data	1. User id 2. In-game shop access
Test Scenario	Verify that if we open the in-game shop and purchase a character for which we do not have enough coins does it give error or not
Step Details	1. Open the game 2. Select the shop option from main menu 3. Purchase a character you do not have enough coins for

Expected Results	1. shop should open 2. Game should not allow you to use that character 3. Error should be given
Actual Results	1. game open 2. shop open 3. Insufficient coins message displayed
Status	1. Pass 2. Pass

Table 6.8: In-game shop fail test case

6.2.2.8 Pause menu Pass

Test Case ID	**PM-001**
Test Case Description	Test the pause menu of the game
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023

Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Game should be started
Test Data	1. User id 2. User name
Test Scenario	Verify that if we pause the game the pause menu appears and from the pause menu choose to go to the main menu does it function properly or not
Step Details	1. Open the game 2. Start the game in single player mode 3. Select main menu option from pause menu screen
Expected Results	1. Game should open 2. pause menu displayed 3. Game should go to main menu

Actual Results	1. Game opened 2. Game started in single player mode 3. After we paused the game the pause menu appeared and from there we went to main menu
Status	1. Pass 2. Pass 3. Pass

Table 6.9: Scenery Change Pass Test Case

6.2.2.9 Pause menu fail

Test Case ID	**PM-002**
Test Case Description	Test the pause menu of the game
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Game should be started

Test Data	1. User id 2. User name
Test Scenario	Verify that if we pause the game the pause menu appears and from the pause menu choose to restart the game does it still open main menu or not
Step Details	1. Open the game 2. Start the game in single player mode 3. choose restart option
Expected Results	1. Game should open 2. pause menu displayed 3. Game restart
Actual Results	1. Game opened 2. Game started in single player mode 3. Game restarted and did not go to the main menu
Status	1. Pass 2. Pass 3. Pass

Table 6.10: Scenery Change Pass Test Case

6.2.2.10 Scenery change pass

Test Case ID	**SC-001**
Test Case Description	Test the scenery change option of the game
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Game should be opened
Test Data	1. User id 2. User name
Test Scenario	Verify that if we select a scenery does it change in the game or not
Step Details	1. Open the game 2. Select the scenery change option 3. Select a scenery
Expected Results	1. Game should open 2. Option should be displayed 3. Change apply

Actual Results	1. Game opened 2. Scenery change option appeared 3. Selected scenery was displayed
Status	1. Pass 2. Pass 3. Pass

Table 6.11: Scenery Change Pass Test Case

6.2.2.11 Scenery change fail

Test Case ID	**SC-002**
Test Case Description	Test the scenery change option of the game
Created By	Talha Shauket
Reviewed By	Fawad Waheed
Version	1.0
Tester's Name	Talha Shauket
Date Tested	10-April-2023
Prerequisites	1. Access to Mobile Phone 2. Game should be installed 3. Game should be opened
Test Data	1. User id 2. User name

Test Scenario	Verify that if we change the scenery of the game the changes apply or not
Step Details	1. Open the game 2. Select the scenery change option 3. Choose night scenery and check if it still opens in day scenery or not
Expected Results	1. Game should open 2. menu displayed 3. Game does not start in day scenery
Actual Results	1. Game opened 2. Scenery change option appeared 3. Game started in night scenery and did not start in day scenery
Status	1. Pass 2. Pass 3. Pass

Table 6.12: Scenery Change Pass Test Case

Chapter 7

Conclusions

In this thesis, we have explored the topic of developing an endless runner game and provided an in-depth analysis of the subject matter. We have discussed the various aspects of game development, including game design, programming and testing, and examined the different phases of game development. Through our research, we have demonstrated the importance and relevance of game development, and provided insights into its impact on the gaming industry and society as a whole. We have highlighted the challenges and opportunities of game development, including the complexity of creating engaging and immersive games, the importance of innovation and creativity, and the potential of games to educate, entertain, and inspire. We have also discussed the best possible approach and techniques used in game development, such as agile development. Additionally, we have explored the different game engines, tools, and technologies used in game development, such as Unity, Blender etc. Overall, our study has contributed to the knowledge and understanding of development of an endless runner game, and we hope that our findings will be useful to game developers, students, researchers, and enthusiasts. We have identified several areas for future research, including the integration of emerging technologies such as virtual reality and artificial intelligence into game development, the impact of games on social issues, and the development of games for education and training purposes. In conclusion, developing any game is a complex and dynamic field that requires a combination of skills, creativity, and technology. Through our research, we have demonstrated the importance and relevance of game development, and provided insights into its challenges and opportunities. We believe that our study provides a valuable contribution to the field of game development, and we hope that it will inspire further research and innovation in this exciting field.

7.1 Future Work

Like any program or anything in the world our game also has a lot of room for improvements. Some of the improvements we will be working on are listed below

1. **Improved Graphics:**
   We will improve our game's graphic as the newer devices have a great framerate and better processing power than older phones so will be make use of this and constantly improve the graphics of our game.

2. **Google Play Store:**
   We are aiming to launch our game on the google play store for this purpose we will make a google developers account and launch our game there.

3. **iOS availability:**
   At present our game is only available on android but we are aiming to make it available for iOS as well in the near future.

4. **Improving multiplayer option:**
   We will constantly improve the multiplayer option of our game to support more and more users and keep the users interacted.

References

[1] C. S. González. *Multiplayer Online Games*, volume 13. 2021. Cited on p. 6.

[2] J. Momoda. Endless runner games: Evolution and future. 19, 2020. Cited on p. 7.

[3] Nitin. Change control by pressing a button using c# script in unity. 18, 2021. Cited on p. 8.

[4] R. Palkar. Enhancing code quality with documentation in c#. pages 21–22, 2023. Cited on p. 9.