

BUNNY'S FORTUNE : AN ANDROID 3D ADVENTURE GAME

MAHMUDUL ISLAM MASUM



BUNNY'S FORTUNE : AN ANDROID 3D ADVENTURE GAME

MAHMUDUL ISLAM MASUM

A report submitted
in partial fulfilment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering

UNIVERSITY OF LIBERAL ARTS BANGLADESH
Dhaka, Bangladesh

October 2021

DECLARATION

I declare that this report entitled “*Bunny’s Fortune: An Android 3D Adeventure Game*” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : 

Name : MAHMUDUL ISLAM MASUM

ID : 171014087

Date : October 10, 2021

CERTIFICATE OF APPROVAL

The project report entitled “*Bunny’s Fortune: An Android 3D Adventure Game*” is submitted to the Department of Computer Science and Engineering at University of Liberal Arts Bangladesh (ULAB) in partial fulfillment of the requirements for the degree of Bachelor of Science.

Dated: October 10, 2021

Prof. Syed Akhter Hossain, PhD

Professor & Head of Department
Computer Science and Engineering
University of Liberal Arts Bangladesh

Signature & Date

Dr. Abul Kalam Al Azad

Associate professor
Computer Science and Engineering
University of Liberal Arts Bangladesh

Signature & Date

DEDICATION

To my lovely parents, who gave me endless love, trust, constant encouragement over the years, and for their prayers.

To my mentors and peers for their patience, support, love, and for enduring the ups and downs during the completion of this project.

This report is dedicated to them.

ACKNOWLEDGEMENT

I wish to express my deepest appreciation to all those who helped me, in one way or another, to complete this project. First and foremost, I thank Almighty who provided me with strength, direction and purpose throughout the project.

Special thanks to my project supervisor Dr. Abul Kalam Al Azad all his patience, guidance and support during the execution of this project. Through his expert guidance, I was able to overcome all the obstacles that I encountered in these enduring three months of my project. In fact, he always gave me immense hope every time I consulted with her over problems relating to my project.

My friends and family...

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	CERTIFICATE OF APPROVAL	iii
	DEDICATION	iv
	ACKNOWLEDGEMENT	v
	TABLE OF CONTENTS	vi
	LIST OF TABLES	ix
	LIST OF FIGURES	x
 1	 INTRODUCTION	 12
	1.1 Background	12
	1.2 Motivation	13
	1.3 Objectives	14
	1.4 Expected outcomes	15
	1.5 Project Management and Finance	16
	1.5.1 Finance	20
	1.6 Report Layout	20
	1.6.1 Section 1: Introduction	21
	1.6.2 Section 2: Background	21
	1.6.3 Section 3: Requirement Specification	21
	1.6.4 Section 4: Design Specification	21
	1.6.5 Section 5: Implementation and Testing	22
	1.6.6 Section 6: Impact on Society, Environment and Sustainability	22
	1.6.7 Section 7: Conclusion and Future Scope	22

2		
	BACKGROUND	23
	2.1 Preliminaries/Terminologies	23
	2.2 Related Works	24
	2.3 Comparative Analysis	26
	2.4 Scope of the problem	32
	2.5 Challenges	34
	2.5.1 Data safety	35
	2.5.2 Optimising the game	35
3		
	REQUIREMENT SPECIFICATION	40
	3.1 Business process modeling	40
	3.2 Requirement collection and analysis	41
	3.2.1 Requirements:	42
	3.2.1.1 Primary requirements	42
	3.2.1.2 Expected requirements	43
	3.2.1.3 Added requirements	43
	3.3 Use Case Modeling and Description	46
	3.4 Logical Data Model	47
	3.5 Design Requirement	48
4		
	DESIGN SPECIFICATION	50
	4.1 Front-end Design	50
	4.2 Back-end Design	52
	4.3 Interaction Design and User Experience (UX)	54
	4.4 Implementation Requirements	59
	4.4.1 Implementing Front-end	59
	4.4.2 Implementing Back-end	60
5		
	IMPLEMENTATION AND TESTING	61

5.1	Implementation of Database	61
5.2	Implementation of Front-end Design	63
5.3	Testing Implementation	69
5.3.1	Features that need to be tested	70
5.4	Test Results and Reports	71
5.4.1	Simulation Test Result	71
5.4.2	Real User Review	76
5.4.2.1	User 1 review:	76
5.4.2.2	User 2 review:	76
5.4.2.3	User 3 review:	77
5.4.2.4	User 4 review:	77
5.4.2.5	User 5 review:	78
5.4.2.6	User 6 review:	79
5.4.2.7	User 7 review:	79
5.4.2.8	User 8 review:	80
5.4.2.9	User 9 review:	80
5.4.2.10	User 10 review:	81
 6		
	IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	82
6.1	Impact on society	82
6.2	Impact on Environment	85
6.3	Ethical Aspects	86
6.4	Sustainability Plan	87
 7		
	CONCLUSION AND FUTURE SCOPE	89
7.1	Discussion and Conclusion	89
7.1.1	The Obstacles	89
7.1.2	The Achievements	90
7.2	Scope for Further Developments	91
	REFERENCE	92

LIST OF TABLES

NO.	TITLE	PAGE
1	Bunny's Fortune vs other popular games	32
2	Differences between game engines	44
3	C# vs C++ in game development	44
4	Test Case 1	71
5	Test Case 2	72
6	Test Case 3	72
7	Test Case 4	72
8	Test Case 5	73
9	Test Case 6	73
10	Test Case 7	73
11	Test Case 8	74
12	Test Case 9	74
13	Test Case 10	75

LIST OF FIGURES

NO.	TITLE	PAGE
1	Gantt Chart	20
2	Game Development cycle	25
3	Genshin Impact : An open world game	27
4	Subway Surfers	28
5	Temple Run	29
6	Leo's Fortune	29
7	Super Mario Run	30
8	Among Us	31
9	Ocean Horn	31
10	Virtual Device's different configs	33
11	Different screen sizes	34
12	Un-optimized Code	36
13	Optimized Code	36
14	Object with LOD	37
15	LOD mechanism	37
16	Occlusion Culling	38
17	Occlusion Culling mechanism	38
18	Performance before Optimization	39
19	Performance after Optimization	39
20	Types of Ads	40
21	High Poly Model	41
22	Low Poly Model	42
23	3D model creation in Blender	45
24	Use Case Model	46
25	Logical Data Model	47
26	Basic Mechanism of the Back-end	48

27	UI design flow	49
28	Front-end design flow	50
29	Game Window design	51
30	Game Controller design	51
31	Back-end mechanism of the gameplay	52
32	Animation controller	53
33	UI model	54
34	Gameplay Model	55
35	Relation between UX and game logic	56
36	Relation between UX and game logic	57
37	Relation between UX and game logic	58
38	Relation between UX and game logic	59
39	Visual representation of Save system	61
40	Visual representation of Load system	61
41	Visual representation of Save-Load system	62
42	Main Menu	63
43	Level Menu	63
44	Settings Menu	64
45	Store Menu	64
46	About Section	65
47	Warning Message	65
48	Loading Screen	66
49	In-game Pause Menu	66
50	In-game Settings	67
51	In-game Warning Message	67
52	Warning Message 2	68
53	Level Finishing window	68
54	Unity Simulation Tool	71
55	Unity Profiler for testing	75
56	Impact & benefits of gaming in society	84

CHAPTER 1

INTRODUCTION

1.1 Background

Video games have been a part of our society and culture for a long time now. Since the 80's video games have been popular among different generations in different forms. From pixelated 2d games to ultra realistic 3d games, it has come a long way. Our way of playing video games has also changed. Special gaming devices like Playstation, Nintendo, Gameboy became popular when gaming became also popular in personal computers. It only got better by the invention of mobile gaming. There were java and symbian games which had a massive collection of both 2d and 3d games. The number of mobile games was huge, not even mentioning the likes of the world famous 'Snake' game. But mobile games always had limitations. The lack of good visuals and limited missions were just a few of them. But it has all changed massively in the last decade after the growing popularity of smartphones. Obviously games were also playable on multimedia phones from the early 20's but in the last decade it has all changed. We also played many games on Nokia phones during our childhood and always wondered how the game works and what is the mechanism behind it. After getting familiar with smartphone games, curiosity only grew. Also the idea of making a game and playing with friends and family always felt like something else. So as a programmer, making a mobile game was inevitable.

The game is called 'Bunny's fortune'. It is made with Unity Game Engine using c#. Unity is very popular and one of the best free game engines available for now. Bunny's fortune is a RPG game for Android mobile platforms. The genre of this 3d game is adventure and fantasy. The game is for everyone but expected players are mostly kids and the age group of 10-15. So the game is specially made in that way. It

is all about the journey of a rabbit called 'Bunny'. He has lost his way home and he is helpless. Gamers have to play the role of that rabbit and have to reach him to his home by taking challenges and avoiding all obstacles. That is why it is a role playing game. It can be also called a 3d platformer game as it has some inspiration from some retro 2d platformer games like Super Mario.

1.2 Motivation

Every year a lot of mobile apps are coming out developed by Bangladeshi app developers. They are developing all sorts of apps. From messaging apps to social networking apps, from entertainment apps to shopping apps, local app development companies and individual developers are developing everything. But compared to the app development sector, the game development sector hasn't evolved much here. Yes, there are a lot of mobile games coming out, but compared to mobile applications, the number of games is low. So I'm looking forward to being a part of the game development sector with a good game.

If we take a close look at the games developed by our local game developers, we will find some sort of similarities between them. In the playstore, a bunch of games developed by our local developers can be found. But strangely enough, if we filter them out, most of the games are mainly from two particular categories, racing games and shooting games. As those are popular genres, most of the developers are trying to make something out of it. Because of that, we are not getting much variation in mobile games. So, I'm developing a game of a different type, an Adventure game. Another thing I have noticed many times in the mobile games developed locally was the lack of a three dimensional (3D) world. Most of them were two dimensional (2D) strategy or puzzle games. So, I have an aspiration to try something challenging, making a 3D game from scratch.

From the experience of developing both android app and game before, comparatively game development seems more challenging and harder. A three dimensional mobile game needs both programming and graphical work. Every single actions and motions inside the game are generated through some line of C#

code at the back end and to visualize those actions and motions, a lot of graphical work is needed for the front end. So, to make a full size 3D game, I have to work on Programming with C# for the back-end and for the front-end, I have to work with 3D modeling, graphics and motion which will improve my skill in various ways. That is a big motivation for me.

This is an animated fantasy adventure game and a big portion of the expected players of this game will be kids, so the game will be somewhat educational. The gameplay will have a navigation system to give the player direction to follow as well as different tasks to complete. So, by playing this game, they will be better at using navigation and taking new challenges while enjoying the animated gameplay. Developing some sort of educational project is another motivation here.

In addition to that, developing a full size game is always a challenging task for an individual developer. But it can massively boost up the skill and knowledge as well as having some possible business value itself. All these factors inspired me to get started with this project.

1.3 Objectives

The main objective of developing this game is to enter the local game development industry. There are still many gaps in the local game development sector and as mentioned earlier, there are various types of games yet to come with three dimensional world space. So, I'm hoping to come up with a game that will give the players some sort of new experience on a mobile platform. As of now, the number of adventure games in the three dimensional world is pretty low on the Android mobile platform. So, I'm hopeful to be a part of this growing game development sector with my own work.

As mentioned earlier, a game requires different types of work to be developed, no matter whether it's a computer or mobile game. As I'm developing the whole game as an individual, I need to do all the programming, 3d modeling, designing, animation, graphical user interface development, and data management work all

myself. Among them, I knew how to do some of them and I had to learn the rest to finish up the project. For example, I really didn't know how to make animations or how to work with models in a three dimensional space before, as all I was able to do was writing lines of codes. But when i started to work for this project i had to learn how to animate objects, how to make 3d models, how to link them up, how to control them with lines of C# code. All these things were challenging to learn and implement practically but it enriched my knowledge. I learnt new tools while it also boosted up my programming skill and that is the biggest objective of this project.

In addition to that, a product like a full sized game also has some business value. While it's boosting up my skills and helping me to learn about new tools, it may also be a good option to do some business in the marketplace if everything goes well. Right now in the Android mobile marketplace (PlayStore), many developers are using their applications and games to generate some revenue. It all depends on various things like the quality, number of users and also a bit of luck sometimes, but it's never out of equation. Having an actual playable game in the marketplace is adding up the possibilities of generating some revenues while developing the whole game from scratch is adding up new knowledge, skills and weight to the portfolio; all at the same time. So, achieving all these were the objectives of this project.

1.4 Expected outcomes

The expected outcome of this project is a full sized 3D open world android RPG game with touch based interactive user interface. After wrapping up the development work, an Android Application package (.apk) file will be built. These are some special files with (.apk) extensions that are executable on android devices. What it does is, containing all the game files in a comperatively compact size. Similarly, I will build the game in that specific format as it's an android game. The expected size of this game is 75-100 mb and it will be playable in both android mobile and tablets. The Apk file will be uploaded in some data hosting site where it can be downloaded easily and later on the marketplace after more testing. The game will have standard low poly style game characters, 3D animated environment, multiple levels with different difficulties. The main game character is a rabbit who

has lost his way home. The player's task is to help him to get to his home. On the path home, there will be many difficulties; including enemies that can attack the player. The player has to collect food items to restore stamina and stay alive. He can also attack the enemies for self defense. There will also be some special abilities that can be collected through picking various items. After completing the level, the next level will be unlocked which will have a harder difficulty level. Players will have the options to adjust the graphics level and audio volume from the setting menu with ease. All the progress and records will be saved locally in the system folder of the device. To ensure security of the saved files, data will be encrypted. Overall, the game will give the players a taste of intense gameplay and moderate graphics on their mobile devices.

1.5 Project Management and Finance

From starting to finishing the project on time needs some effective planning and management. This project is being developed by following the standard software development life cycle. The software development life cycle has 7 crucial phases:

- i. Planning
- ii. Analysis
- iii. Design
- iv. Development
- v. Implementation
- vi. Testing
- vii. Maintenance.

Apart from the regular steps of SDLC, game development process has 7 crucial stages:

- i. Planning
- ii. Pre-production
- iii. Production

- iv. Testing
- v. Pre-launch
- vi. Launch
- vii. Post-production

1. Planning

Before the authors start composing, the planners start planning, and the developers start their work, a plan for the game needs to surface [4]. This is the absolute initial segment of the arranging stage and the roots that each computer or mobile game will develop from.

In this stage, the most essential inquiries should be addressed, as:

- i. What kind of game is it?
- ii. Will it be 2D or 3D?
- iii. What are some of the key features it should have?
- iv. Who are its characters?
- v. When and where does it occur?
- vi. Who is our main target user group?
- vii. Which platform will be used?

It may not seem like it, yet ideating a game is perhaps the hardest part. A proof of idea takes every concept that has been created and perceives how suitable they are for the gaming studio to deliver. From that point, extra inquiries should be responded to, as:

- i. What will be the estimated cost?
- ii. Do we have the capacity to develop it?
- iii. What is the estimated time frame for development to launch?

2. Pre-production

The following phase is called pre-production, conceptualizes how to offer life to the numerous thoughts spread out in the planning stage.

3. Production

During this phase:

- i. Character models are planned, delivered, and iterated on to look precisely how they should appear in the story.
- ii. Level designs are mapped.
- iii. Programming a huge number of lines-of-source code to bring each piece of in-game objects to life.

4. Testing

Each element and features in the game should be tried for quality control. A game that hasn't been completely tried is a game that is not prepared for an Alpha delivery. Here are a few things a playtester may test during this stage:

- i. Are there buggy scenes or levels?
- ii. Is everything delivered to the screen?
- iii. Are there any hacks to take advantage of the game?

5. Pre-launch

This stage is an unpleasant time for developers. Developers can't help thinking about how the public will respond to the first release of the game.

6. Launch

At this stage, the actual game gets released. Normally these builds have a lot of bugs and glitches that get fixed with further updates. This phase is also important for making proper SEO which is crucial for ranking the game higher on PlayStore. The better the rank gets, the more popular the game becomes. This is also the best time for promoting and marketing the game on different platforms.

7. Post-launch

Post-launch is one of the most interesting steps in game development. It's normal for games to dispatch with bunches of minor bugs. The initial few months during this stage additionally depend on players to submit bug reports or make some noise about bugs in web-based platforms. This is all important for post-launch support. One more part of this stage is to give regular updates to the game. These updates range from game-adjusting patches to new downloadable features.

All these phases had to be completed within 14 weeks. Planning the story, gameplay, analysing the requirements and preparing them took almost 3 weeks. The implementation of features and all the back-end work was done in the following 7 weeks. The visual enhancements and game art were added in the following 2 weeks. Building the first prototype and testing was done in the last 2 weeks. Here is a Gantt chart of the development time frame.

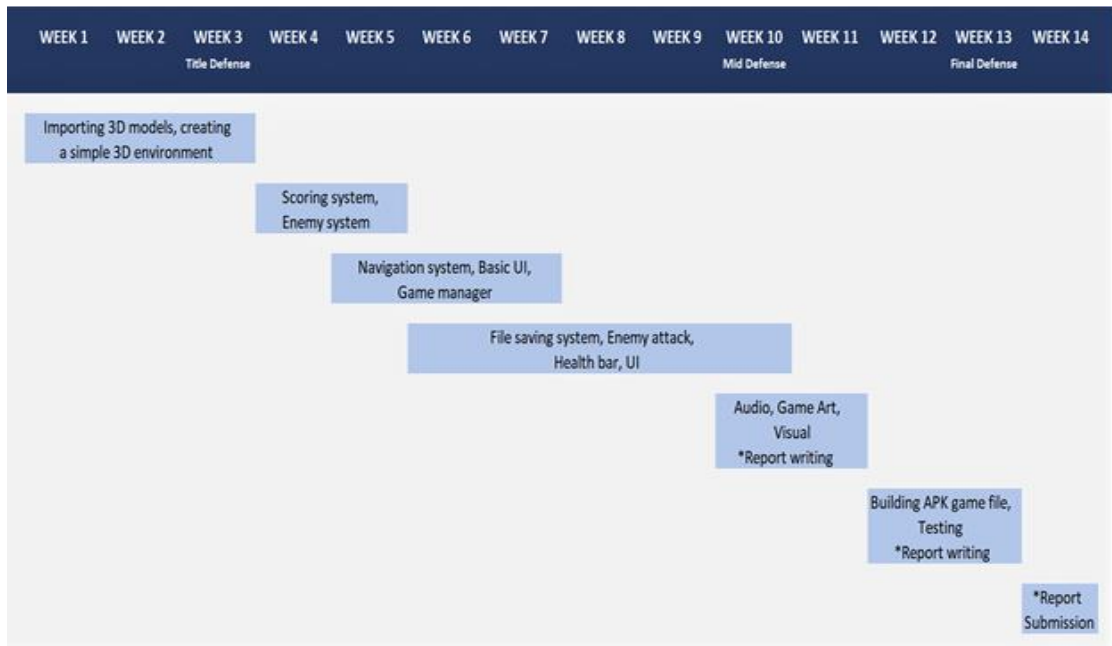


Figure 1: Gantt Chart

1.5.1 Finance

All the game characters, models, animations and UI elements in this game were made from scratch to avoid copyright issues. Later a few copyright free unity assets, music, sprites and images were used to enhance the visual and overall experience of the game. The development of this project did not cost any funds as all the important elements used were self-made from scratch.

1.6 Report Layout

This report is divided into seven main sections. Each section has several sub-sections and it also includes the approval, declaration, acknowledgments, list of figures, list of tables and references.

1.6.1 Section 1: Introduction

In this section, the project idea, its background, and motivations behind the project is discussed. Also what are the objectives, what is expected as the outcome, how the project was managed, what was the budget and also the project planning, scheduling and timetables were discussed in this chapter.

1.6.2 Section 2: Background

In this section, the game is compared with other popular games out there to discuss the difference and inspirations regarding the project. There was a lot of conceptual work and brainstorming was needed as well as analysing the requirements. All these are mentioned in this section. Also during the development, what were the challenges and what can be the problem in the future is also discussed.

1.6.3 Section 3: Requirement Specification

This section is about the requirement analysis and making the game mechanism. An application also needs to have a proper business model and that is also mentioned here. The basic game mechanism and work flow is also shown in the use case model and logical data model.

1.6.4 Section 4: Design Specification

This section is about one of the most important phases in software development, designing. Both the Front-end and the Back-end are discussed here with simple graphs. Apart from the Interaction Design and User Experience models are briefly discussed. The required software and hardware list for developing and playing this game is also mentioned in this section of the report.

1.6.5 Section 5: Implementation and Testing

This part of the report talks about the implementation and outcomes of this project. From this chapter, we know about how the database works and how the front-end design interacts. Also we can know about the testing process and reports and reviews from simulation tests and real users respectively.

1.6.6 Section 6: Impact on Society, Environment and Sustainability

This section is about the ethical aspects of the outcome of this project. Every software has some impact on society and the environment. This chapter is all about that. Also the long term sustainability plan for the game is discussed here.

1.6.7 Section 7: Conclusion and Future Scope

This is the conclusion section of the report. The overall impact of the project is discussed here and scope for further development is mentioned.

Apart from these, the report also includes the approval, declaration, acknowledgments, list of figures, list of tables and references.

CHAPTER 2

BACKGROUND

2.1 Preliminaries/Terminologies

Scripting: Refers to coding or programming. It is the way of writing instructions using a specific programming language that executes all the actions of a game.

The scripting language of this game is C#

Low Poly: Three dimensional models made using a low amount of polygons.

Polygon: A computer generated series of lines used to create a 3d object.

2.5D: A visual way of portraying a three dimensional world space within a two dimensional setup.

RPG: Role Playing game.

Sprite: Bitmap images.

Physics engine: Premade feature that replicates real-life laws of physics in a game.

Unity assets: Characters, objects, sound effects, maps, environments, etc.

Cinematics: Piece of animation that depicts the story of the game.

Culling: Detection and rejection of unnecessary data.

Lightmap: A pre-rendered lighting system.

Mesh: A group of computer generated vertices, edges, faces and possibly with texture.

Fps: Frames per second.

Prefab: Modified gameobjects.

LOD: Level of details of an object.

Shaders: A program that controls light and shadow effects.

Visual scripting: An unique way of generating codes visually by playing with some visual blocks of codes.

Runtime: A rendered state.

Texture mapping: Applying textures to game assets.

Profiling: Monitoring the memory usage of the game for optimization.

Post processing: Special image effects that enhance the visuals.

2.2 Related Works

In game development, brainstorming is an important factor to pull up new ideas.

Apart from the phases of standard software development life cycle, some other steps also need to be performed. Here are some important related work:

- i. Concept / idea generation
- ii. Brainstorming
- iii. Prototyping
- iv. Play test
- v. Implementation

Concept or idea generation is the very first step of any type of game development. This is the stage when developers compare and study other games just to bring an innovative or unique idea.

Then to improve those ideas, a lot of brainstorming is necessary because it helps the developer to make working prototypes of every single new feature or idea.

Prototypes then go through several testing to be selected for the final implementation. This is a repetitive and continuous process and often performed with the actual development phases of SDLC simultaneously.

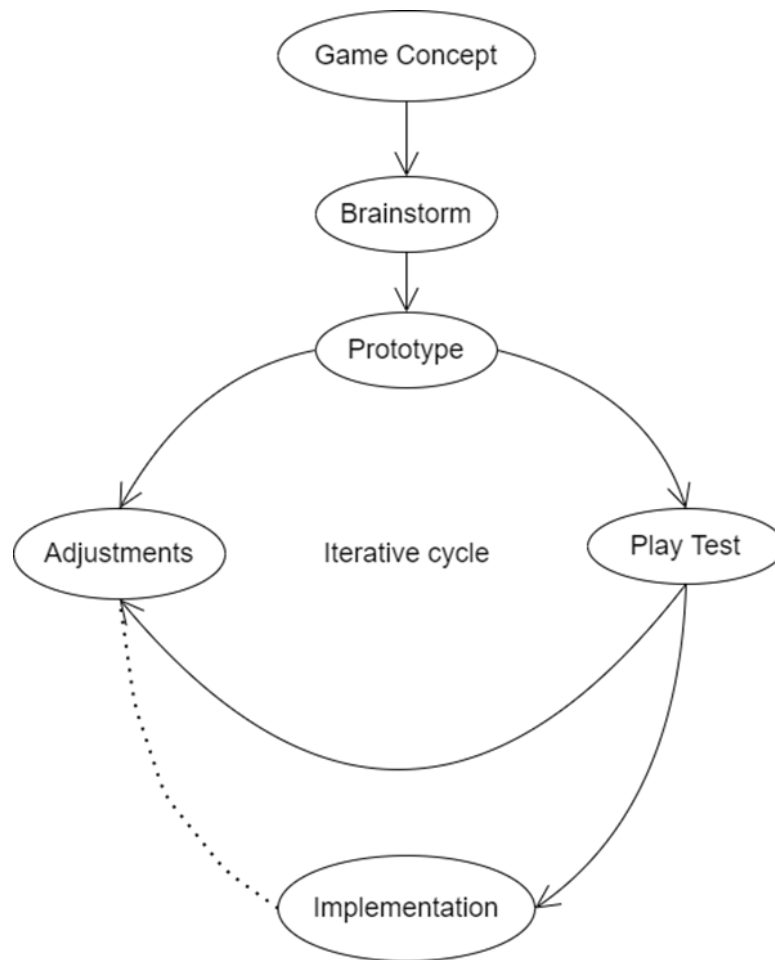


Figure 2: Game Development cycle

In this case, a lot of other games were played to study the features, control, and gameplay. Game reviews on the PlayStore were also studied and noted thoroughly. It was essential to know about the preference of the players and users. In the quickly developing field of computer programming and advancement and surprisingly more quickly developing areas of game improvement, what's to come is difficult to anticipate. Success can be estimated by investigating the subsequent programming. In a game task, the item is a game. Be that as it may, and here comes the point: A game is significantly more than simply its product. It needs to give content to become agreeable. Very much like a web developer: without content the developer is pointless, and the quality can't be estimated. This importantly affects the game task all in all. Some portion of the undertaking isn't the one to focus on, and it

should be considered in association with any remaining parts: The atmosphere of the game, the story, characters, game plays, the craftsmanship, and so on.

There are also some differences between C# for software or web development and for game development. So, it also needed to be learnt as well as some 3d modeling work which was almost a new thing because all the previous games were made using free assets from the asset store.

2.3 Comparative Analysis

There are many adventure games now available on the PlayStore. They come with their own features, story and gameplay. But there are some basic features and parameters that are vital for a game to be successful nowadays.

- i. Gameplay
- ii. story
- iii. world space
- iv. multiplayer features
- v. open world surfing abilities
- vi. pricing
- vii. size

Gameplay refers to the story and control of the game which is the most important part of any game. It plays a massive role in a player's game playing experience.

World space refers to the dimension of the game environment. It can be two dimensional or three. Sometimes it can also be 2.5D which is a special illusional effect that fakes the three dimensional space within a two dimensional setup.

Multiplayer features normally suggest if the game has the option to create, host and join a room where other players are also interacting in real time.

Open world surfing is an important and crucial part of a game these days. Most of the games have a limited path for playr to follow whereas open world games give the player the ability to roam around freely.



Figure 3: Genshin Impact : An open world game

Pricing gives us the idea if the game is paid or free. Sometimes a game is free but has ingame item purchase features to generate revenue.

Game size refers to the disk space taken by the application after getting installed.

Here is a comparison of the game “Bunny’s Fortune” with six popular android adventure games right now,

- i. Subway Surfers
- ii. Temple Run
- iii. Leo’s fortune
- iv. Super Mario Run
- v. Among Us
- vi. Ocean Horn

This games are specifcily choosen because they are proven successful in recent years and met all seven parameters mentioned above.

Game name: Subway surfers

Initial release date: May 23, 2012

Genres: Endless Runner

Mode: Single Player video game

Engine: Unity

Developers: Kiloo, SYBO Games

Platforms: Android , iOS, Windows, macOS

Google Play rating: 4.5/5

Kiloo.com rating: 4.6/5

App Store rating: 4.6/5



Figure 4: Subway Surfers

Game name: Temple Run

Initial release date: August 3, 2011

Genres: Endless Runner

Mode: Single Player video game

Developers: Imangi Studios

Platforms: Android , iOS

Google Play rating: 4.1/5

App Store rating: 4.4/5



Figure 5: Temple Run

Game name: Leo's Fortune

Initial release date: April 23, 2014

Genres: Platformer, Adventure

Mode: Single Player video game

Developers: 1337 & Senri LLC, Tilting Point

Platforms: Android , iOS, Windows, macOS, PlayStation 4, Xbox one

Google Play rating: 4.7/5

App Store rating: 4.7/5



Figure 6: Leo's Fortune

Game name: Super Mario Run

Initial release date: Decemeber 15, 2016

Genres: Platformer

Mode: Single Player video game

Engine: Unity

Developers: Nintendo

Platforms: Android , iOS

Google Play rating: 3.8/5

App Store rating: 3.8/5



Figure 7: Super Mario Run

Game name: Among Us

Initial release date: June 15, 2018

Genres: Platformer, Adventure

Mode: Multi Player video game

Engine: Unity

Developers: InnerSloth LLC

Platforms: Android , iOS, Windows, PlayStation 4 etc

Google Play rating: 3.6/5

App Store rating: 4.2/5



Figure 8: Among Us

Game name: Oceanhorn

Initial release date: November 14, 2013

Genres: RPG

Mode: Multi Player video game

Developers: Cornfox & Brothers Ltd.

Platforms: Android , iOS, Windows, PlayStation 4 etc

Google Play rating: 3.9/5

App Store rating: 4.3/5



Figure 9: Ocean Horn

Table 1: Bunny's Fortune vs other popular games

Game Name	World Space	Multiplayer	Open World	Price	Size
Subway Surfers	3D	No	No	Free	126 mb
Temple Run	3D	No	No	Free	47 mb
Leo's fortune	2.5D	No	No	Paid	250+ mb
Super Mario Run	2D	No	No	Free	200+ mb
Among Us	2D	Yes	No	Free	125 mb
Ocean Horn	3D	No	No	Free	800 mb
Bunny's Fortune	3D	No	Yes	Free	65 mb

One particular thing has to be noted that this is a 3D platformer game for Android devices. We have countless numbers of 2D platformer games on PlayStore but the number of platformer games with three dimensional world space is still not that much. On AppStore, wonderful 3D platformer games like 'Hot Lava' are available for Apple iPhone devices but those are yet to come to Android devices.

2.4 Scope of the problem

Since the arrival of the first iPhone, touch based mobile games are becoming popular day by day. As of now, there are millions of games on both Google PlayStore and Apple AppStore and many of them have millions or billions of users. There are all sorts of games available including action, RPG, racing, driving, adventure, puzzle, sports games. Hundreds or thousands of new games are getting released everyday so the competition is intense. Ranking a game higher is now

dependent on so many things. There are many good games that still remain unnoticed thus becoming unsuccessful. So only the gameplay or story is not enough, rather the exposure and publicity play a huge role in the success of a game.

Music is a vital part of any game. It is essential for a full sized game. Perfect music can easily tell us about what is going on at a game level. It makes the game more intense, lively, and engaging. A game needs different types of sounds. But for developers or programmers, it's not always easy to produce or make good music for their game. Downloaded music does not always go well with the gameplay and it may also have copyright issues. So managing and adding music was a problem but eventually was solved using some copyright free music.

Another problem was testing the game on different devices. As Android phones come with different configuration and screen sizes, the performance needs to be continuously checked to keep a balance between the visual quality and gameplay performance. So it was fixed using some virtual machine softwares like MemuPlay to test the game using different screen ratio and processing power.

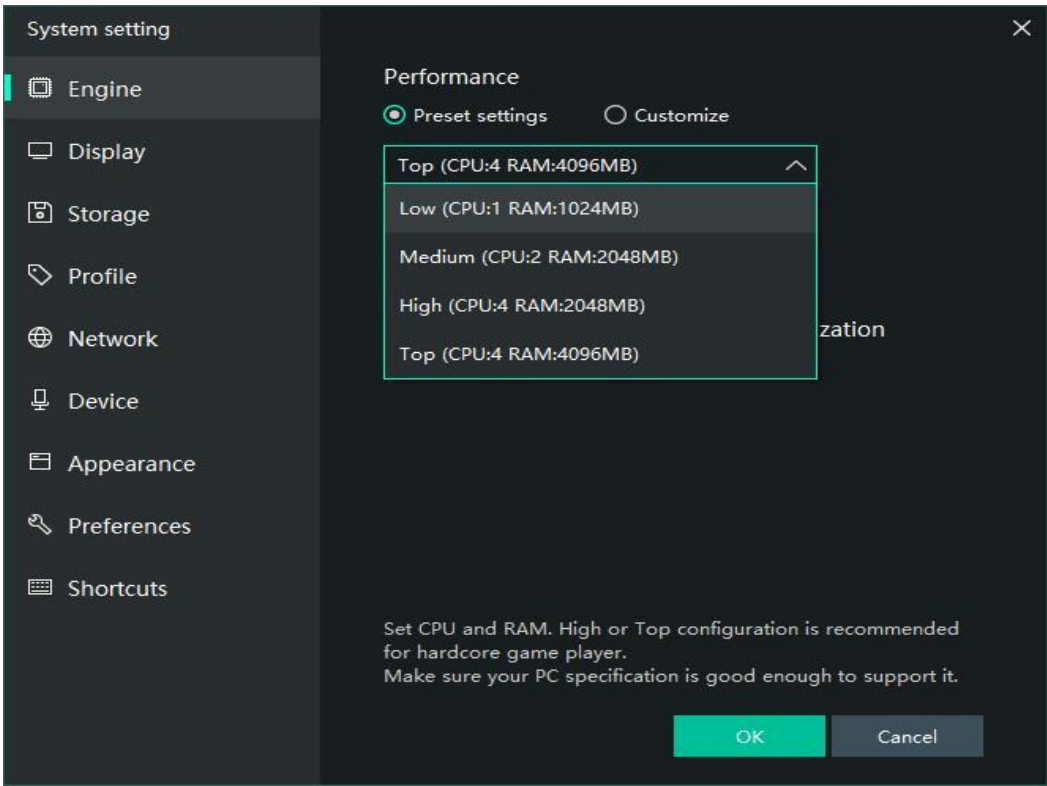


Figure 10: Virtual Device’s different configs

2.5 Challenges

From the technical and development aspect, the biggest challenge for a 3d mobile game is the optimisation issue. All the Android phones are not equally powerful. The main difference between IOS and Android devices is versatility. There are entry level, mid level and flagship Android phones with different configurations and the game has to be optimised for all of them. The developer has to adjust it accordingly and that's the biggest challenge in mobile game development.

Another common issue is with the screen ratio. As there are hundreds of different Android phones with different screen sizes, it has to come with an option of changing screen ratio dynamically based on the specific phone. Sometimes, it can be really problematic if not fixed.

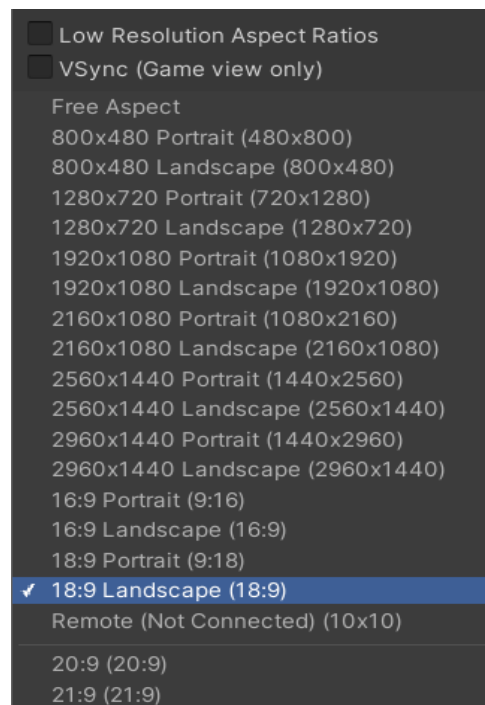


Figure 11: Different screen sizes

2.5.1 Data safety

Every game has its own data saving system. Some games use a server while others use the local system. This particular game is using a local system to store the game progress, settings and every other important data. Local file saving system comes with its own pros and cons. The data can be read and written faster as it does not require any internet connection. But on the other hand, it is also less secure than storing something on a remote server. So, to prevent this, the data is going through 'Binary encryption' which will prevent users from modifying or hacking game data.

2.5.2 Optimising the game

As mentioned earlier, optimising the game is the biggest challenge in game development and there are many ways to achieve the better spot between looks and performance. The game has to look good while it will also perform well. To achieve this, the most common way is to reduce graphics quality but that will also hamper the visuals. That's why optimised Low-Poly models were used to reduce memory load.

Something that is often overlooked is 'Script optimisation'. The performance of a game is heavily dependent on the lines of the codes a developer wrote. Sometimes, unnecessary or obsolete lines of code can decrease the performance rapidly. For example, unity has two types of main functions. One is the 'Start' function and the other one is the 'Update' function. The Start function only gets called at the first frame during runtime and the Update function gets called at each frame thus 60 times a second. If the programmer put a block of code inside the Update function which was supposed to be inside the Start function, it will get called 60 times every second unnecessarily. The gameplay may just work fine but the cpu will be used unnecessarily which will deduce the fps eventually after a while. So, Script optimisation is vital for getting a consistent fps.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class example : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10     }
11 }
12
13 // Update is called once per frame
14 void Update()
15 {
16     Debug.Log("This is wrong code");
17 }
18 }
```

Figure 12: Un-optimized Code



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class example : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10         Debug.Log("This is right code");
11     }
12
13 // Update is called once per frame
14 void Update()
15 {
16 }
17 }
18 }
```

Figure 13: Optimized Code

Another technique is to use LOD groups. This is basically a group of the same model with different Polygon count. For example, the LOD1 object has 1000 polygons while LOD2 and LOD3 objects have 500 and 300 polygons respectively. The idea is, while the camera is near LOD1, it will render and show the object with 1000 polygons and as soon as the camera goes away from the object, it will render LOD2 and LOD3 depending on a specific distance [6]. That's how the system will reduce the polygon count dynamically which will help to increase fps.

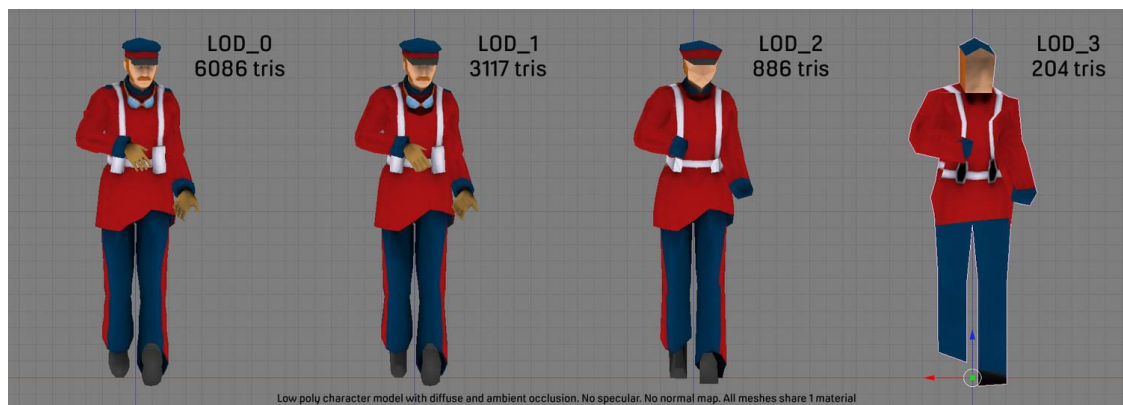


Figure 14: Object with LOD

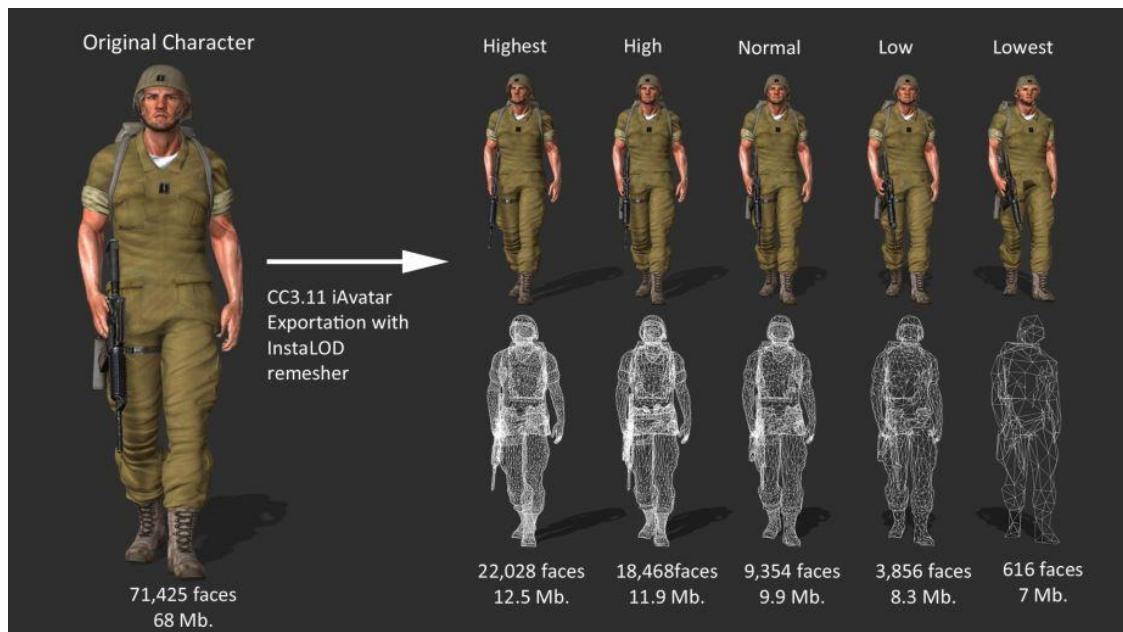


Figure 15: LOD mechanism

Another common way is called ‘Occlusion Culling’. What it does is pretty simple but very effective [5]. It stops rendering the whole scene every time, instead it renders only the objects in front of the camera.

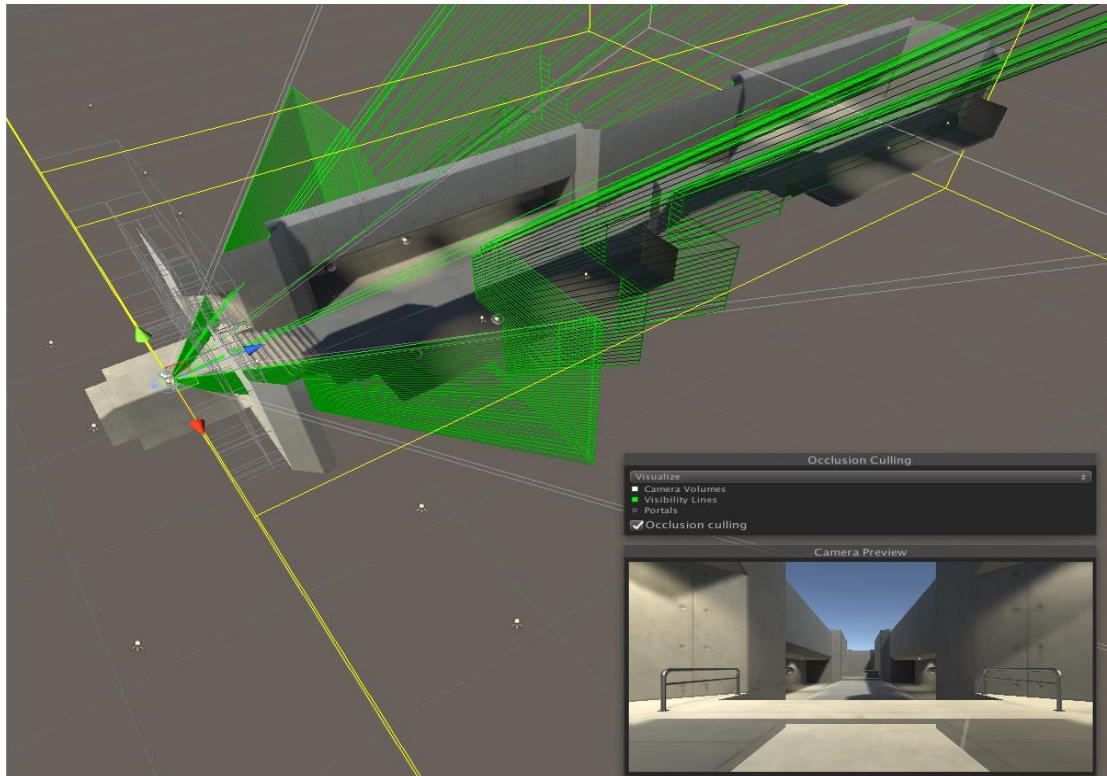


Figure 16: Occlusion Culling

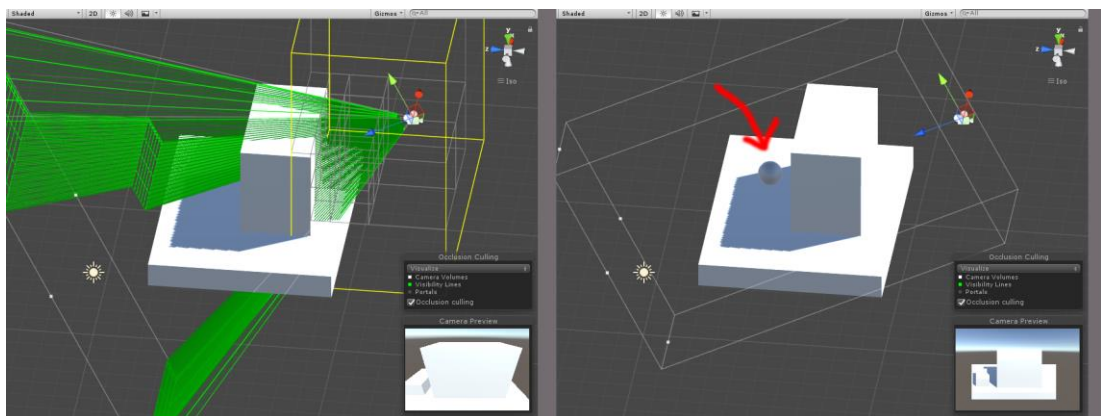


Figure 17: Occlusion Culling mechanism

Unity comes with a pretty useful tool called Profiler. It shows the cpu-gpu usage, data and memory usage with colorful and detailed graphs live during the runtime.

It's really useful to know from graphs what is behind the laggy performance and developers can fix them accordingly.

After optimising, tweaking and adjusting graphics level and scripts, a massive improvement in fps and overall performance can be achieved.

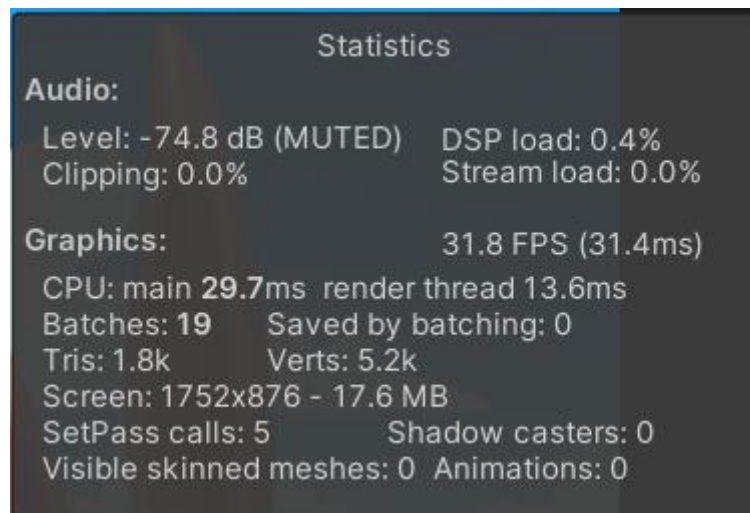


Figure 18: Performance before Optimization

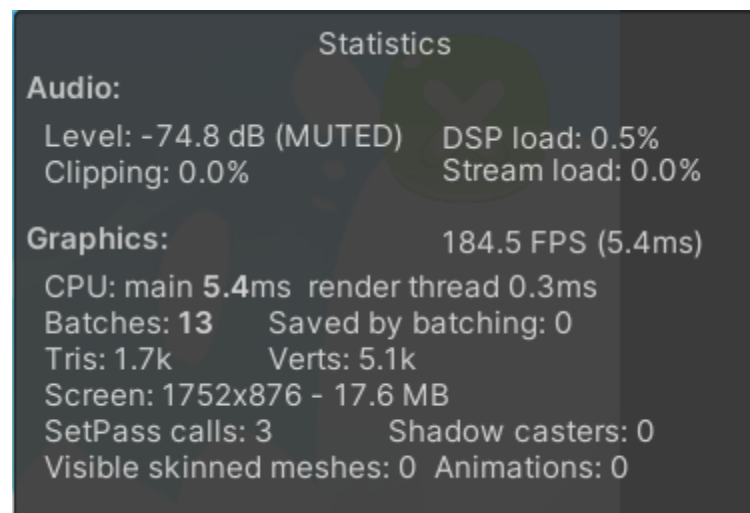


Figure 19: Performance after Optimization

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business process modeling

It's a totally free to play game. After further optimizing, testing and enhancing the visuals even more, it will be available on the PlayStore where anyone will be able to download it for free. However, I will monetize the game with Google Ads so there will be some advertisements in some parts of the game. It's a pretty popular practice. There are different types of 'Ads' available depending on use case. Some 'Banner ads' can be used anywhere on the user interface. There are 'Skippable video ads' or 'Interstitial ads' that can be used whenever the player completes a level. Also, there are unskippable 'Rewarded video ads' which can be used to reward players with some special game items or extra chances. So, the players will use those ads themselves to play and finish the game which will be less annoying for them while having the game for free.

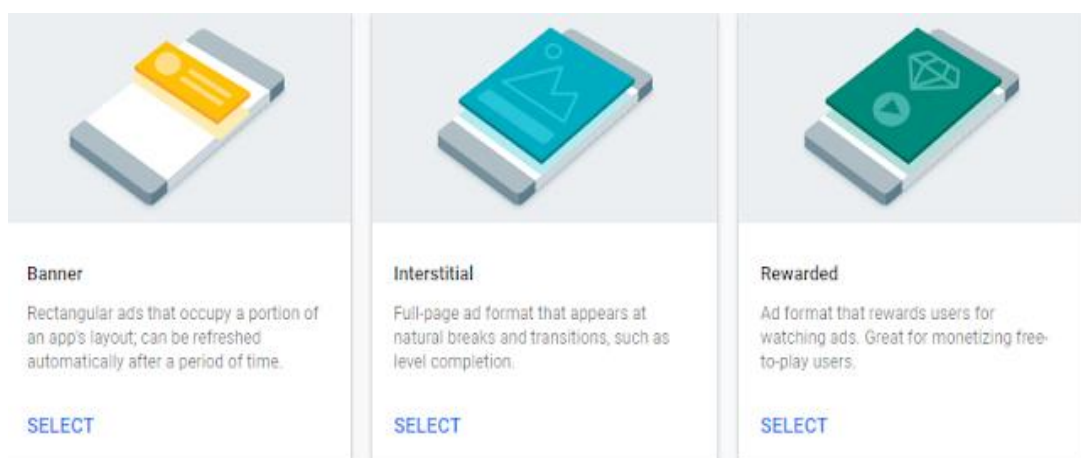


Figure 20: Types of Ads

A ‘Buy the game / stop watching ads’ option will be featured in the game with later updates that will let the user buy the game and have some special abilities to use. Another feature that will come in the further updates is an ingame shop where players can buy some game items.

3.2 Requirement collection and analysis

Mobile devices are less powerful compared to laptops or desktops. They come with less powerful cpu and gpu. So, if a mobile game contains three dimensional world space, it has to be well optimized while having standard or moderate graphics. Also, as the game is being developed by an individual developer within a limited amount of time, it’s tough to pull up ‘PUBG’ level graphics. That's where the ‘low poly’ style models became useful. Every three dimensional model is basically a group of polygons [7]. Low-poly 3d models are a special type of them with way less poly count thus being more optimized for mobile devices by taking less space and render time. Another reason behind going for low-poly style models is, they have an animated cartoonish look into them which is more suitable for the theme of the game.



Figure 21: High Poly Model



Figure 22: Low Poly Model

3.2.1 Requirements:

Three main types of requirements in game development are:

- i. Primary requirements
- ii. Expected requirements
- iii. Added requirements

3.2.1.1 Primary requirements

- i. User friendly and lucrative system
- ii. Minimum maintenance cost
- iii. Ease of update
- iv. Ease of gameplay

- v. Lucrative Story
- vi. Professional thinking
- vii. Business model

3.2.1.2 Expected requirements

- i. Developing project within the budget
- ii. Maximum quality
- iii. Efficient UI
- iv. Optimised gameplay

3.2.1.3 Added requirements

- i. Providing cheat codes
- ii. Running on minimum configuration
- iii. Multiplayer features

An important factor was to choose the right game engine for the mobile game. There are two popular free and widely used engines right now, Unity and Unreal Engine. Unreal Engine uses C++ as scripting language whereas Unity uses C#. Unreal Engine offers better graphics to go with but when it comes to mobile games, there is nothing better than Unity yet. Games made with Unity are smaller in size and there are a lot of free modules and assets to use that are well optimised for mobile devices. So, the Unity game engine was the better option for this game.

Table 2: Differences between game engines

Unity Engine	Unreal Engine
Better for individual developers, coders.	Better for big studios, artists.
Coding can be done using C#, Bolt.	Coding is done using C++, Blueprints.
Better optimised for mobile games.	Better optimised for PC games.
Monetisation on mobile platform is supported.	Not supported.
Smaller game size.	Bigger game size.
Performs well on a medium configuration PC.	Needs high configuration computers.

Another reason for choosing Unity was to use C# as the programming language as it is easier to use compared to C++.

Table 3: C# vs C++ in game development

C#	C++
Easier Memory management.	Has to be done manually mostly.
Works on popular OS.	Does not have any preferred platform.
Easier Garbage collection.	Has to be done manually mostly.
Beginner friendly for game development.	Needs some extra work and adjustment.
More popular for mobile games.	More popular among pc games.

To create the 3d landscapes, levels and characters, a lot of 3d models were needed. It is easy to use some pre-made models from the marketplace but that needs some really good budget and may also create some copyright issues. So, every single 3d model was made from scratch using 3d modeling softwares like blender and Sculpt+.



Figure 23: 3D model creation in Blender

Animations are a major part of any game and they make them more lively. A game needs player's movement animations like walking or jumping, enemy's attack animation, item collection, etc. Some special cinematics are played in some parts of the game. Unity by default comes with some ready to use animations but they are pretty common in various games because of the availability. So, a lot of other animations had to be created. Unity offers some basic animation creation tools but those are not really enough. That's why softwares like Blender and Mixamo had to be used.

Implementing the features within the time is also important and that's why a storyline and plan has to be made.

3.3 Use Case Modeling and Description

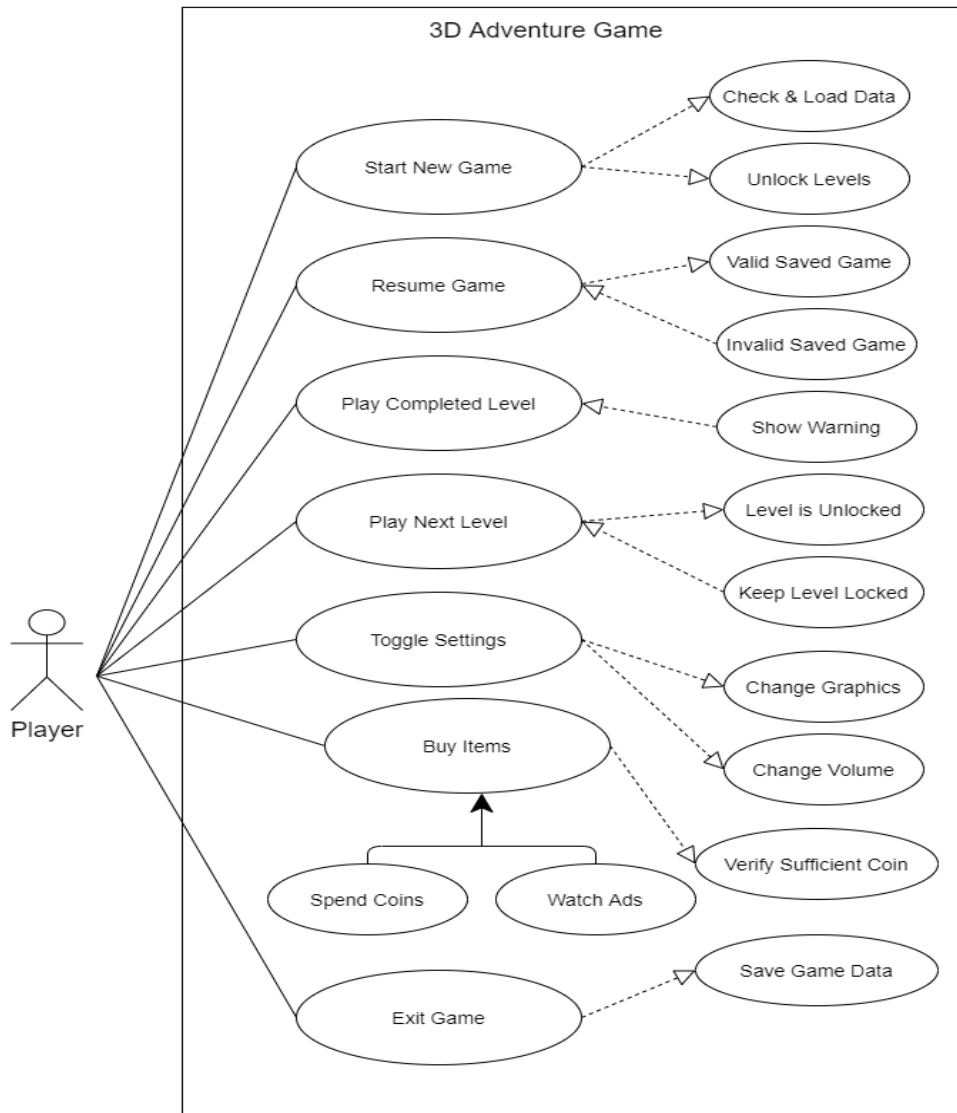


Figure 24: Use Case Model

It is a single player game with various use cases. The user will get the option to start or resume a game depending on the saved game availability. If there is any valid saved game file, the player will be taken to the previously played mission, if there is none, then the player will be taken to the new mission. If the player completes a level or mission, the next mission will be unlocked and the progress will be saved. In the level selection menu, the saved game data will be checked to verify whether the level was actually unlocked. From the settings menu, players can easily change the graphics preset and music volume. In the store, players can buy items with coins

or by watching ads. During purchase, the available coin will be checked whether the player has sufficient coins. In each step, the progress data will be saved and all the data will be loaded the next time the player opens the game.

3.4 Logical Data Model

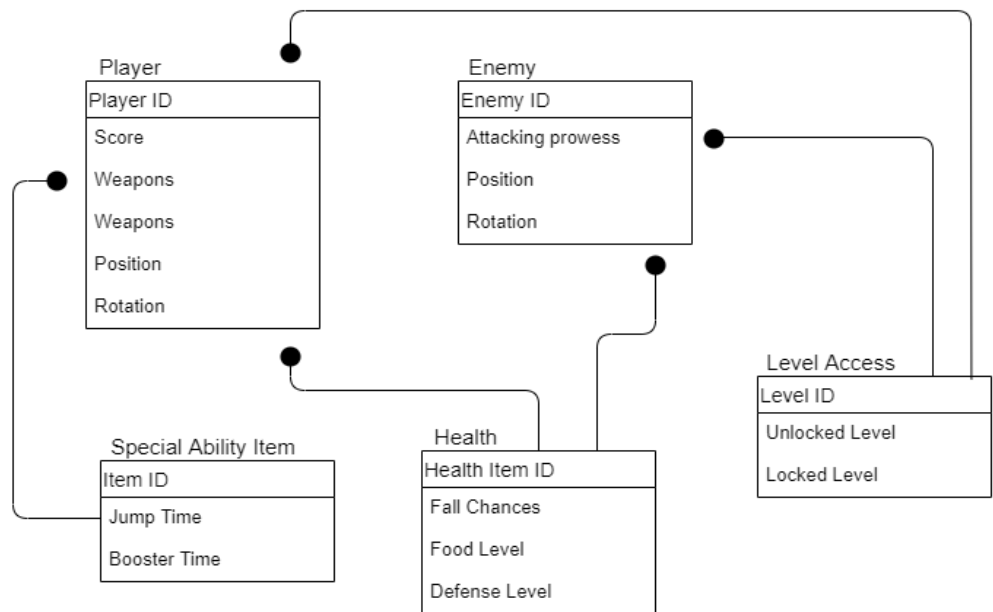


Figure 25: Logical Data Model

3.5 Design Requirement

In any software development, designing is one of the most important phases in Software development life cycle. It is an essential phase. In this phase, developers determine what the software needs, how the visual will be and also the timeline of development. The main purpose is to transform the pre analysed requirements into a proper, complete and detailed system development specifications.

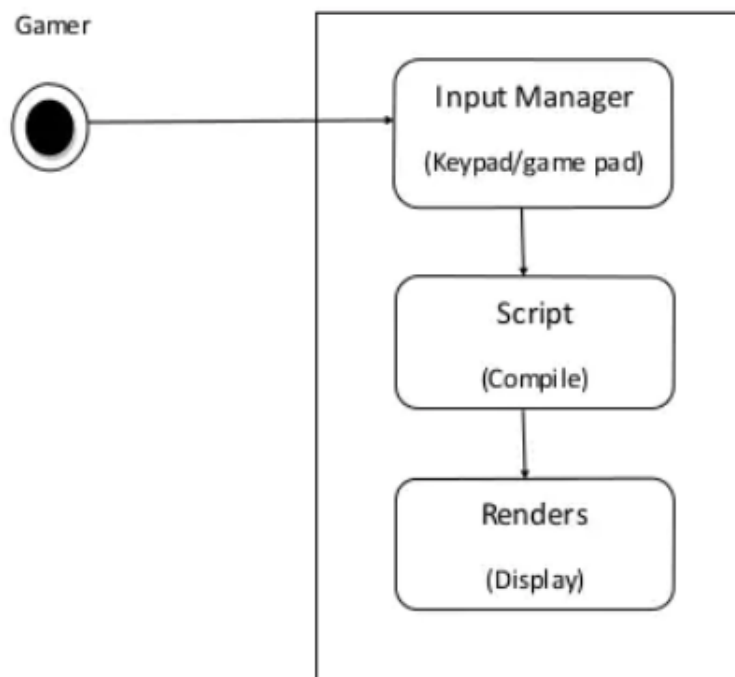


Figure 26: Basic Mechanism of the Back-end

These are the functional attributes that tell the developer what ideas need to be converted to feature. When it comes to development of a game, the first thing that needs to be considered is the wants and needs of the player. This particular game has an expected user group of 8-16. So, all the game level design, UI design, and item design should meet their taste or choice.

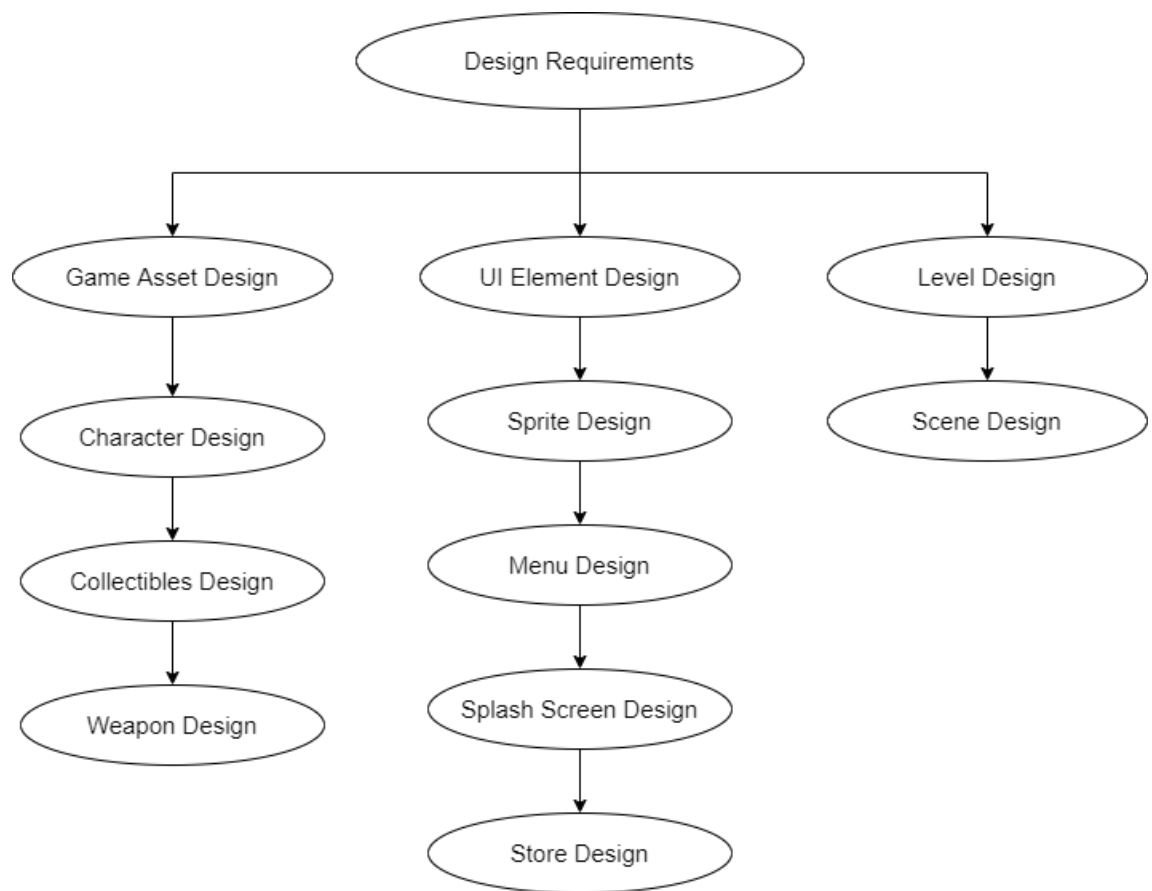


Figure 27: UI design flow

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

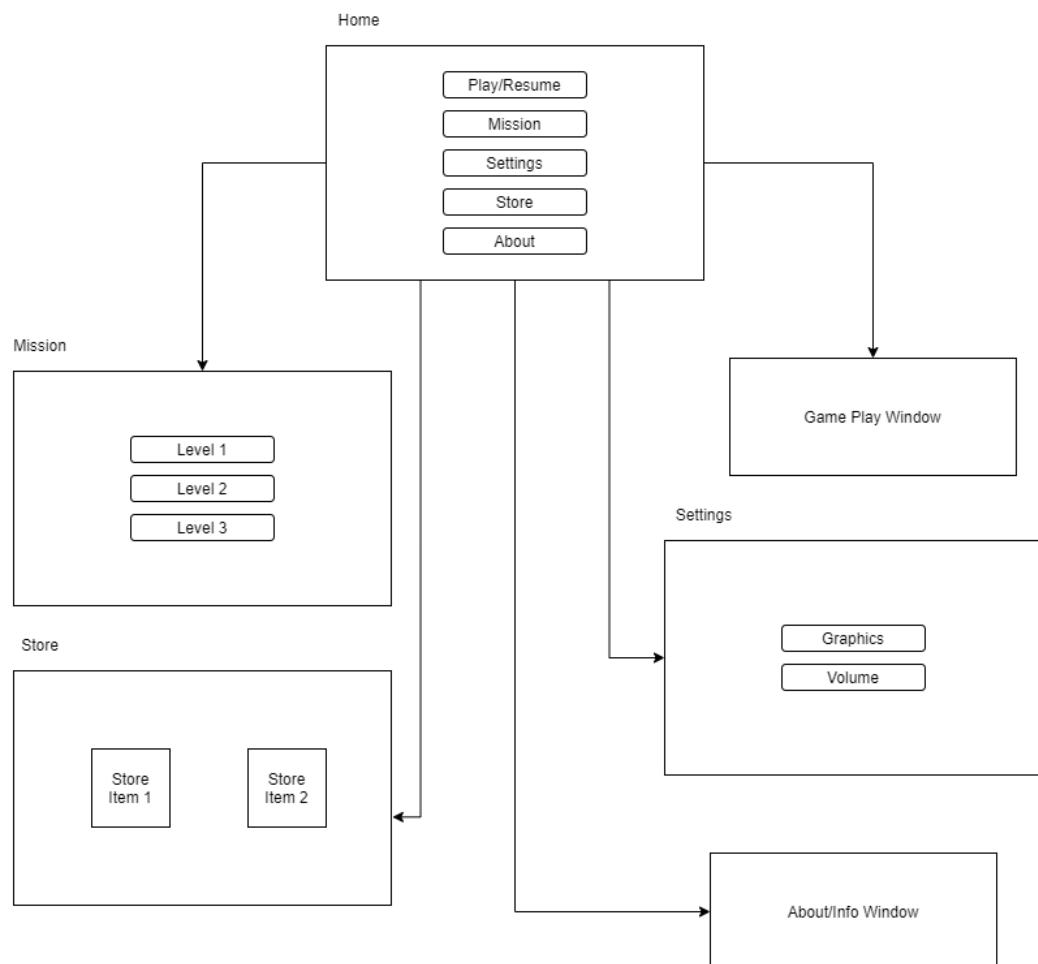


Figure 28: Front-end design flow

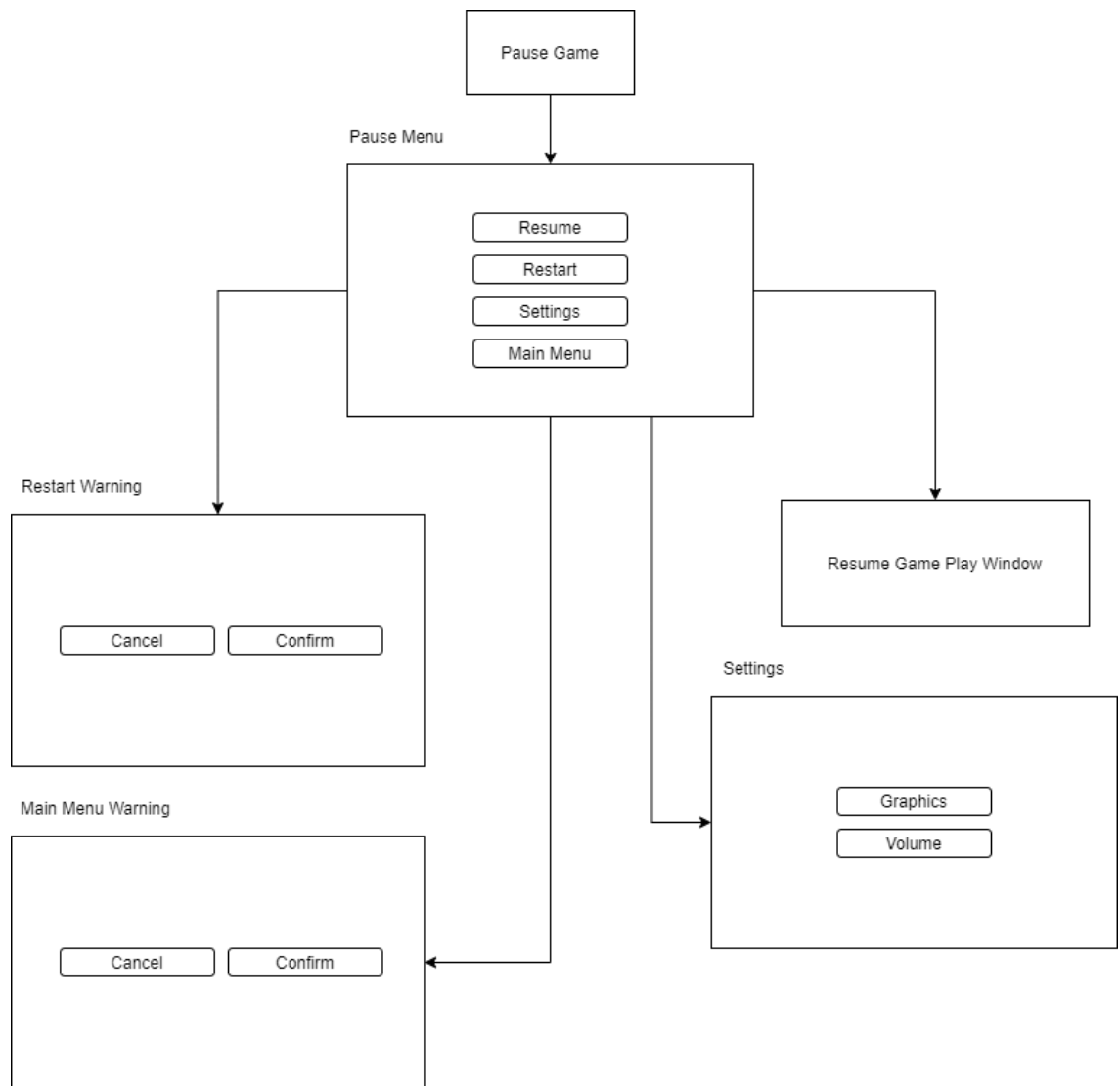


Figure 29: Game Window design

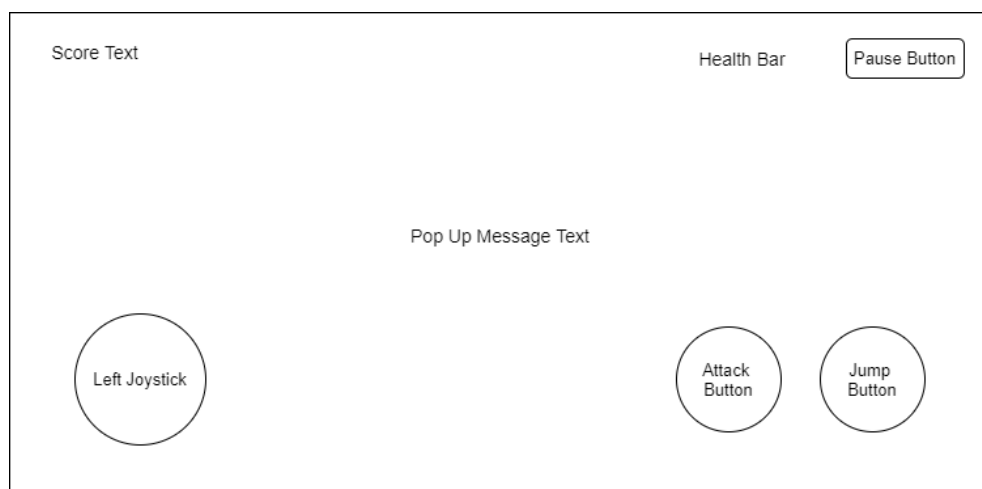


Figure 30: Game Controller design

4.2 Back-end Design

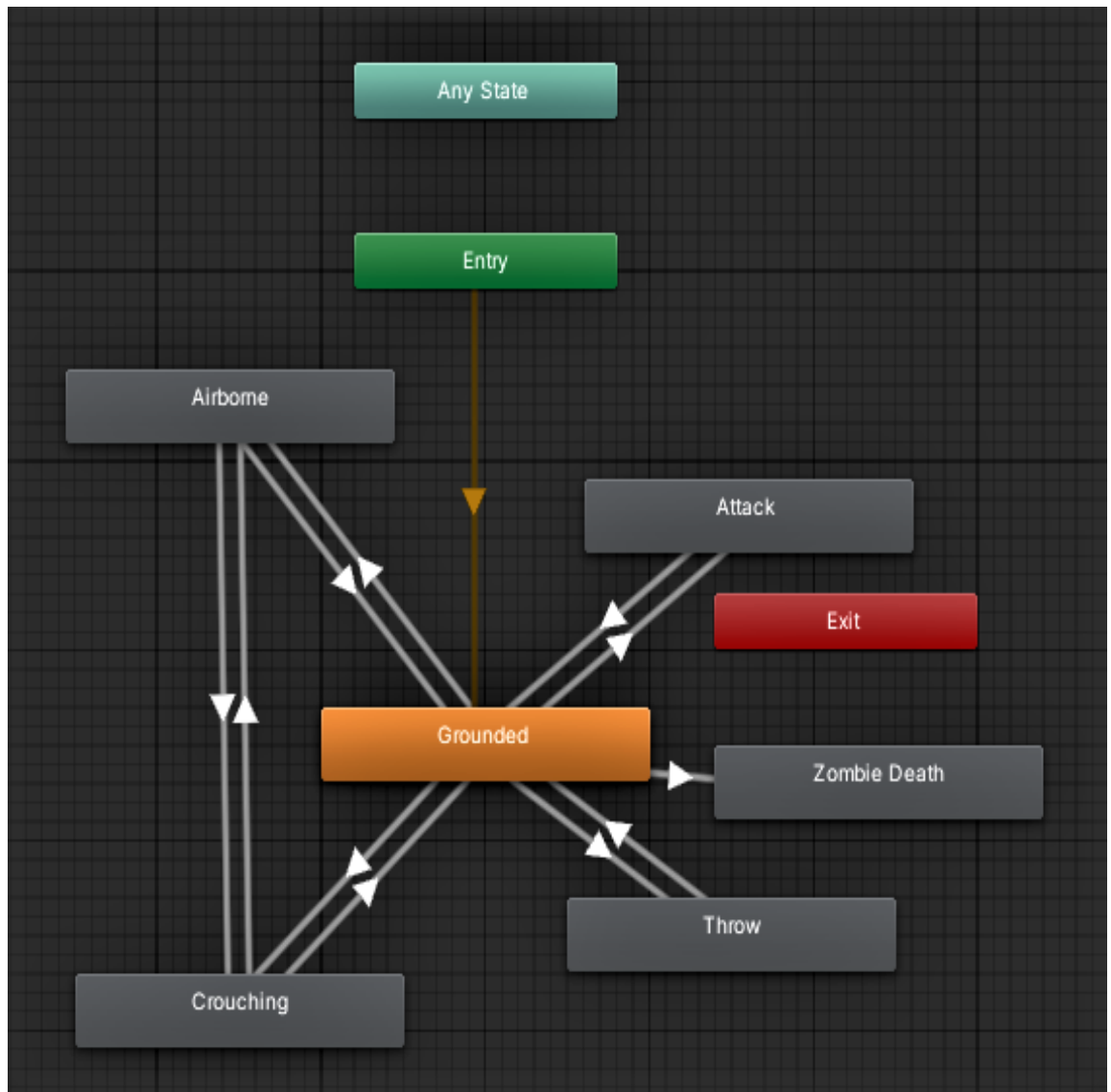


Figure 31: Back-end mechanism of the gameplay

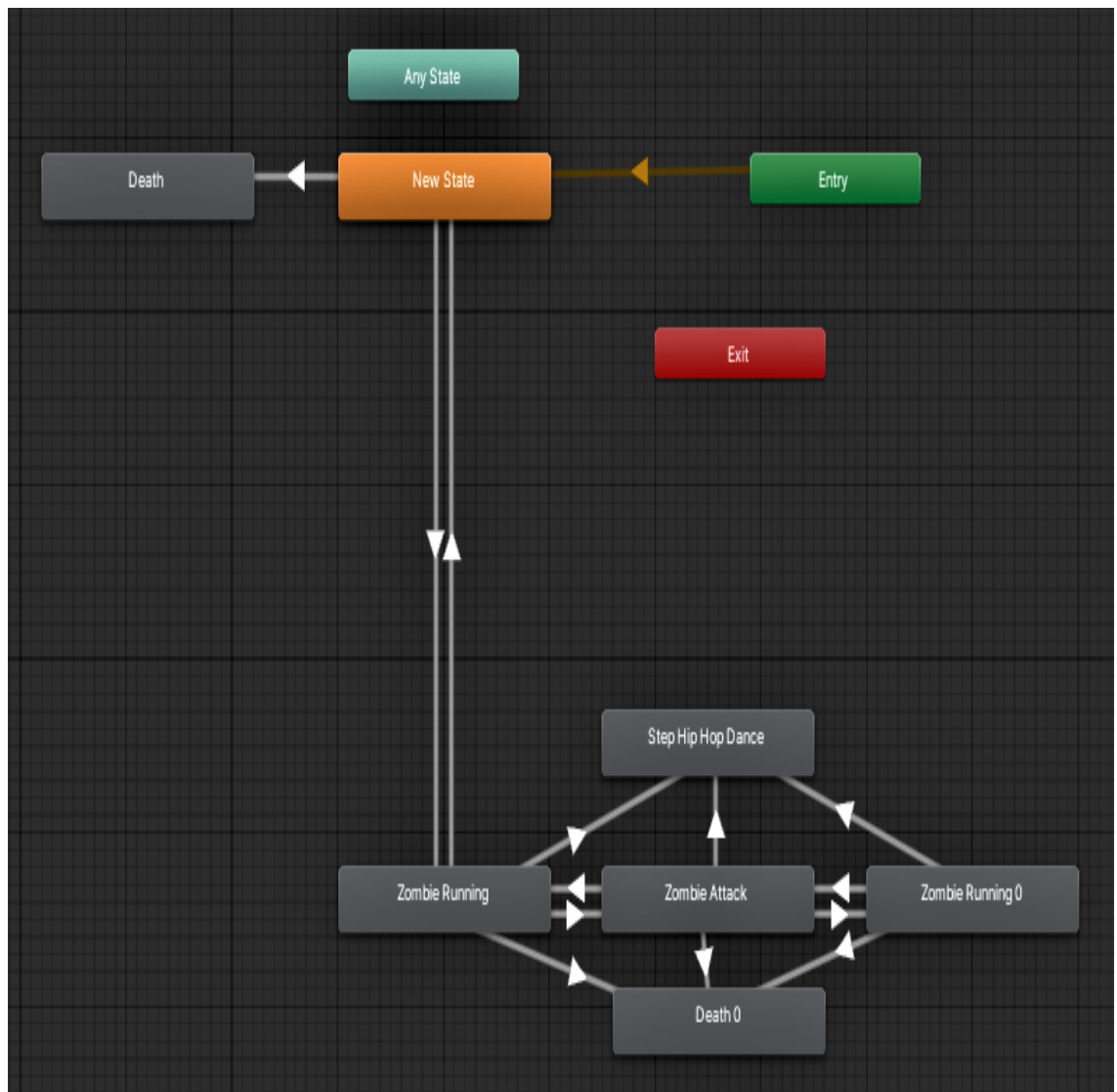


Figure 32: Animation controller

4.3 Interaction Design and User Experience (UX)

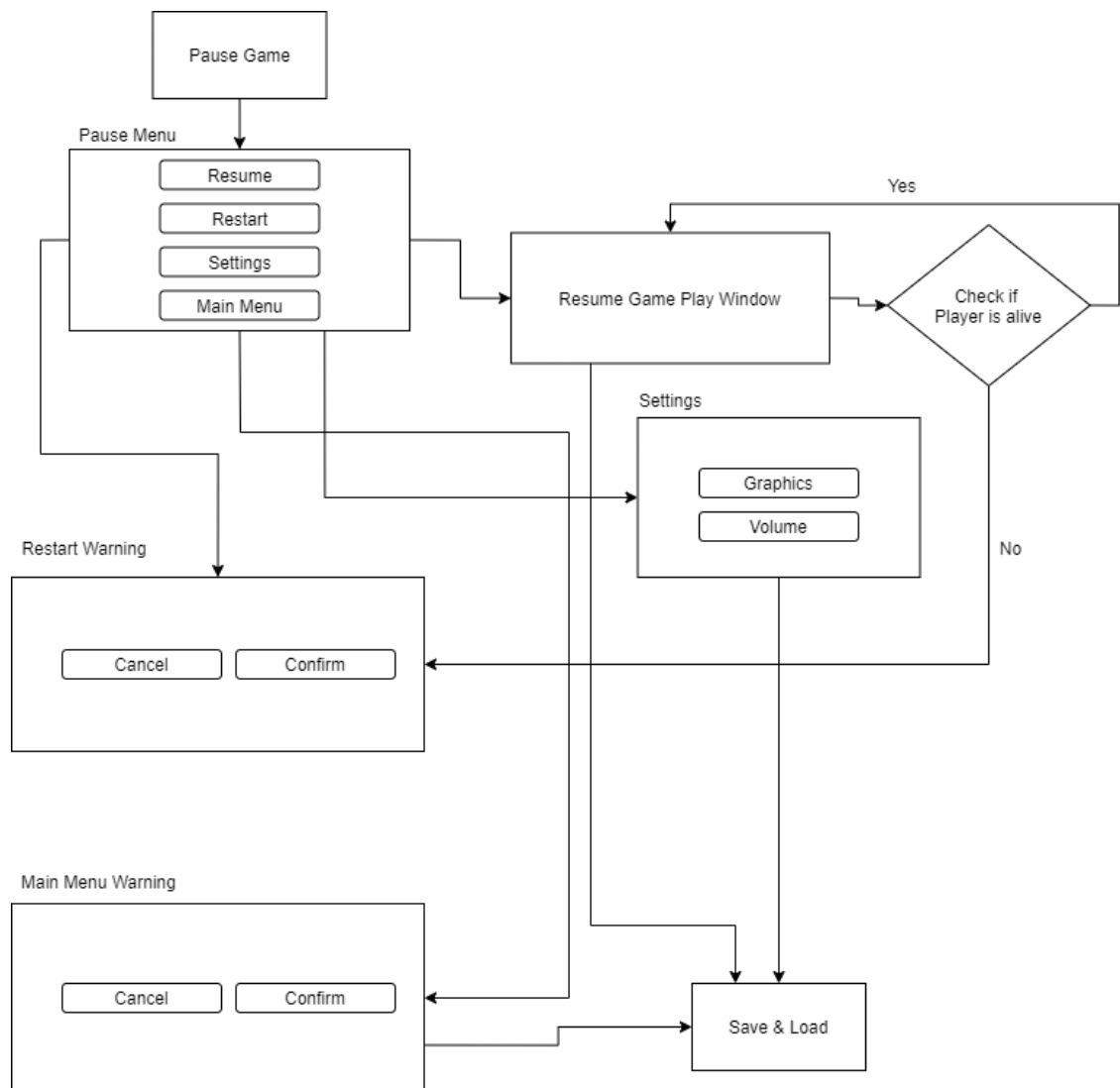


Figure 33: UI model

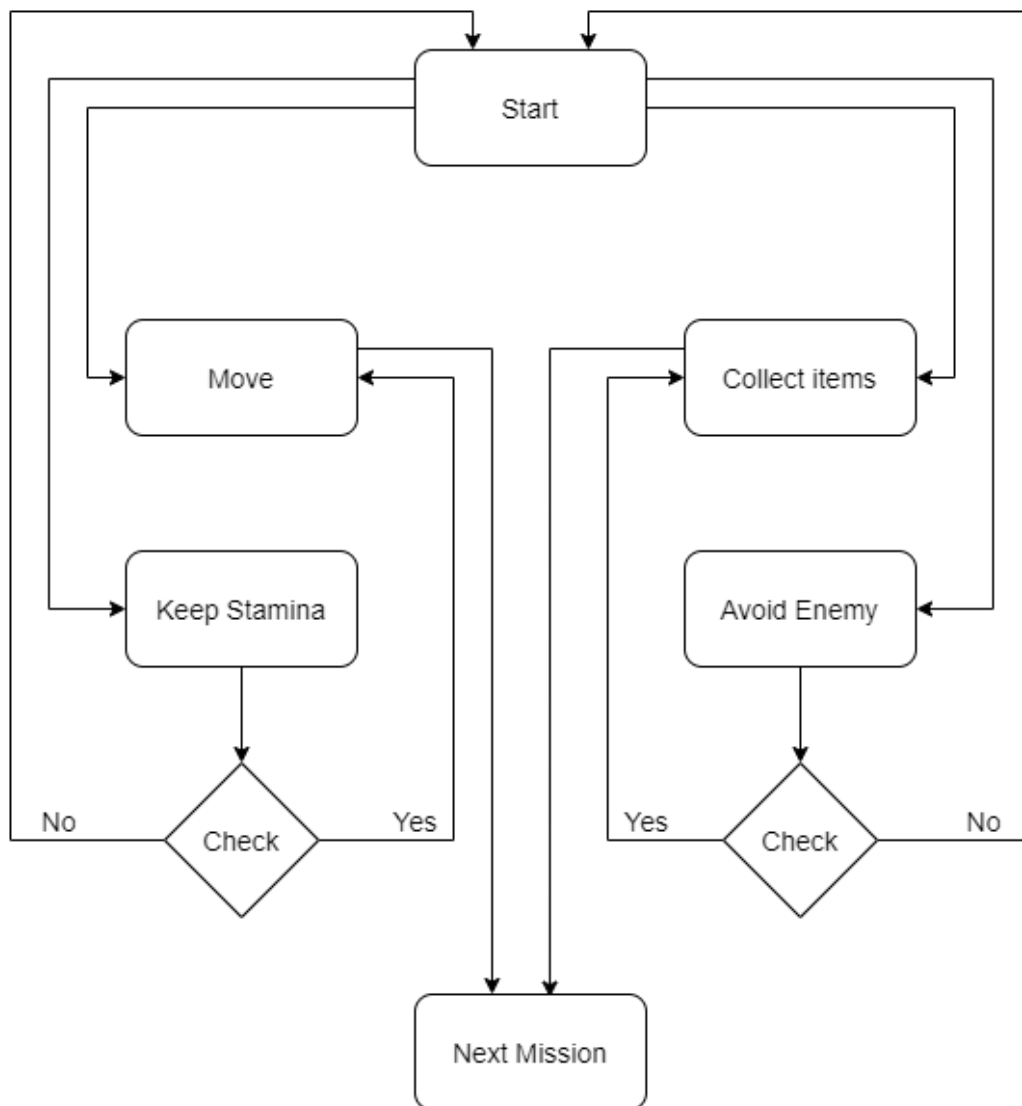


Figure 34: Gameplay Model

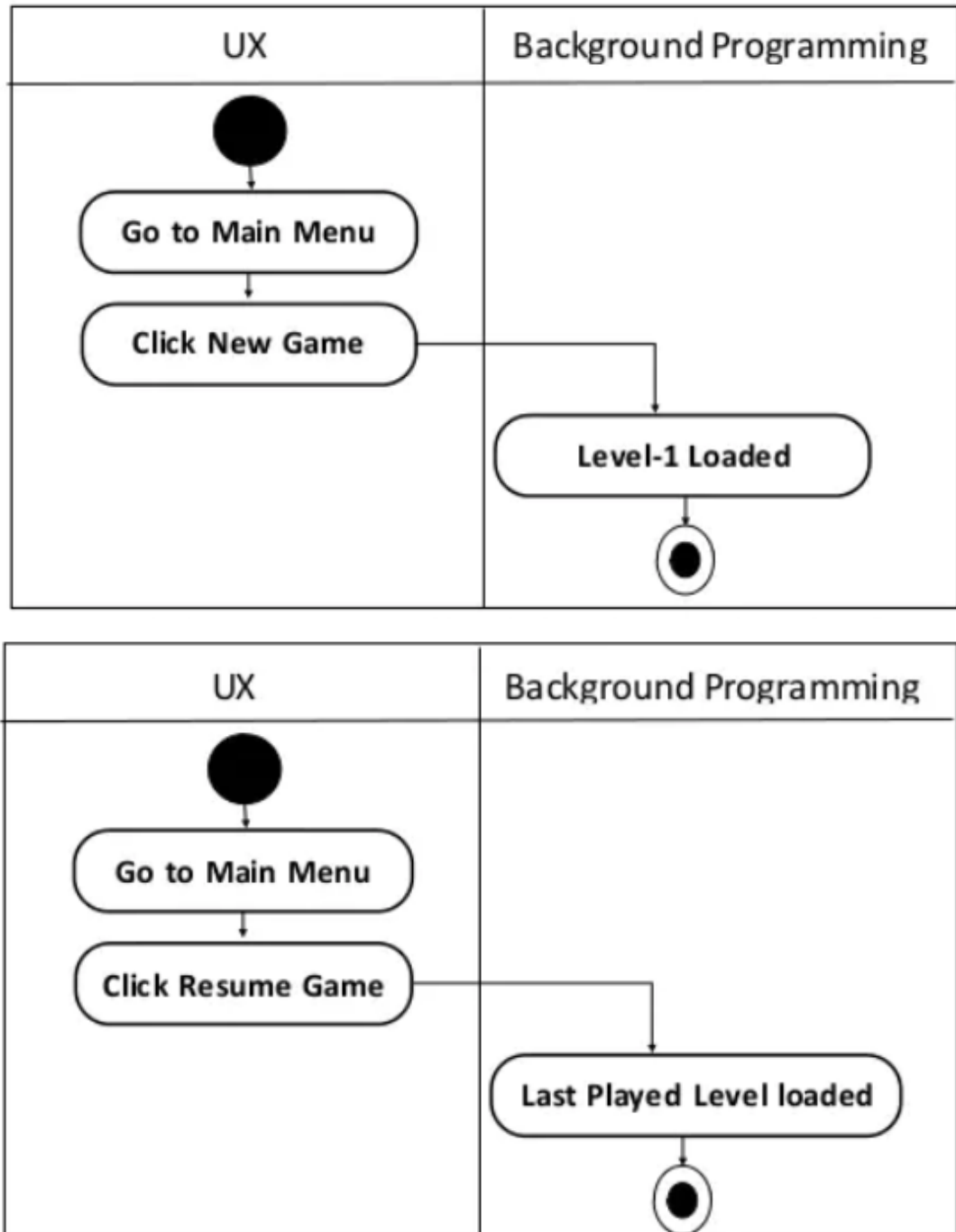


Figure 35: Relation between UX and game logic

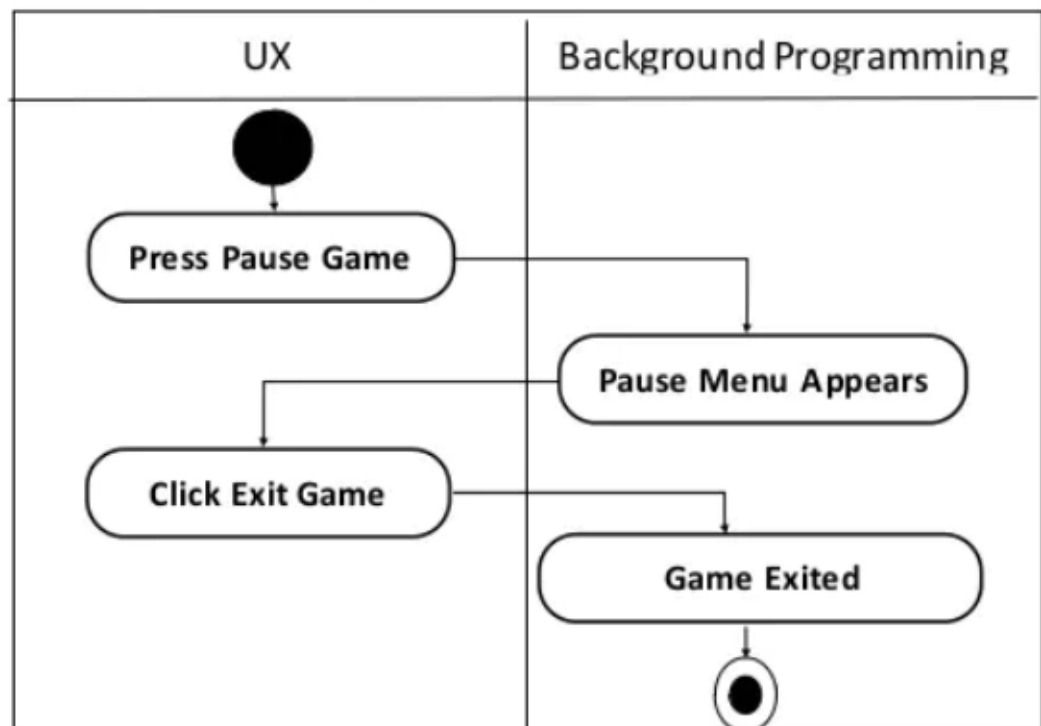
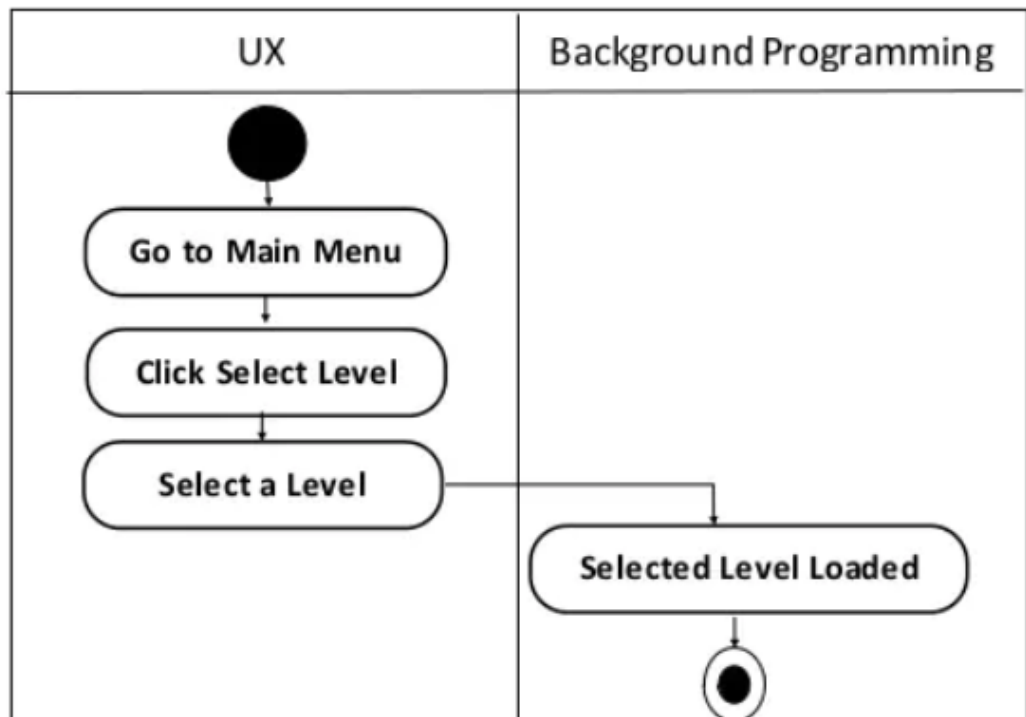


Figure 36: Relation between UX and game logic

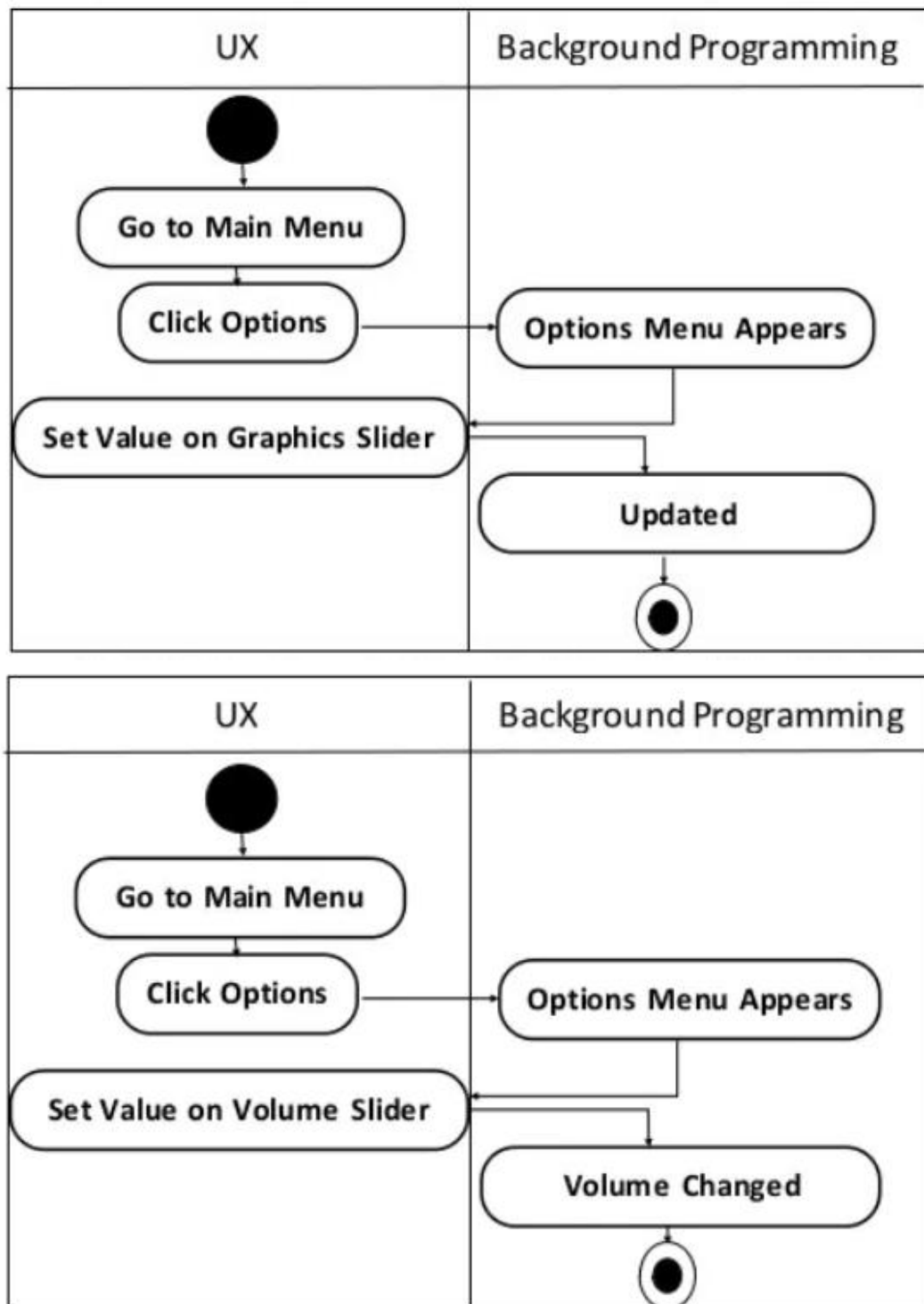


Figure 37: Relation between UX and game logic

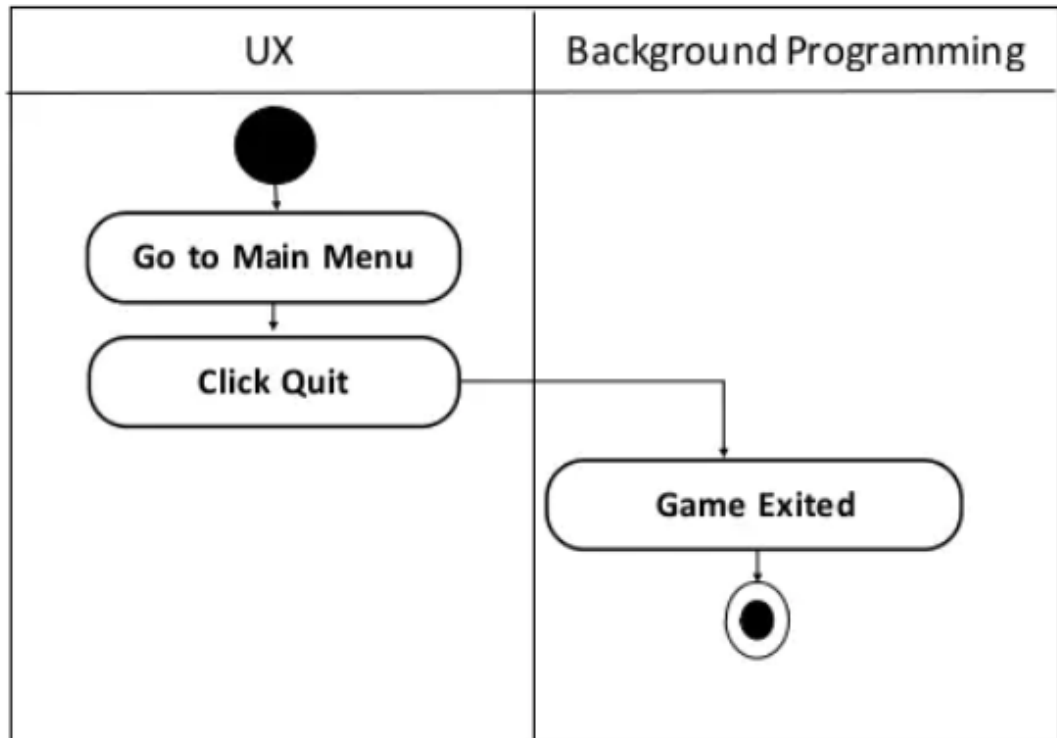


Figure 38: Relation between UX and game logic

4.4 Implementation Requirements

To implement and execute the features, some capable software and hardware is needed. Here is a list of them.

4.4.1 Implementing Front-end

Following software tools were used:

- i. Unity Game Engine
- ii. Android Software Development Kit (Android SDK)
- iii. Blender
- iv. Sculpt+

- v. Mixamo
- vi. Nomad Sculpt
- vii. Adobe Photoshop 2020

4.4.2 Implementing Back-end

Following software tools were used:

- i. Visual Studio 2017
- ii. Visual Studio Code

During development, all these softwares were running in a Computer with following specifications:

- i. Processor: AMD Ryzen 3 3250U
- ii. Ram: 8gb
- iii. Video ram: 2gb Integrated Vega 3
- iv. OS: Windows 10 64 bit

System requirements to run the game smoothly on Android device:

- i. Processor: Snapdragon 625, Mediatek Helio g35
- ii. Ram: 3gb
- iii. Android version: Android 5.0 Lollipop or up
- iv. GPU: Adreno 506 or equivalent

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

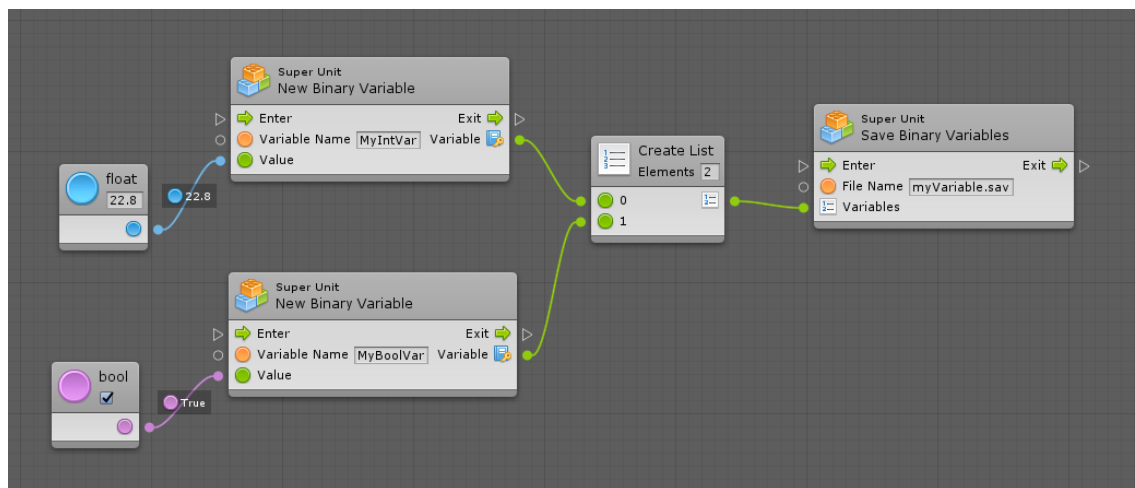


Figure 39: Visual representation of Save system

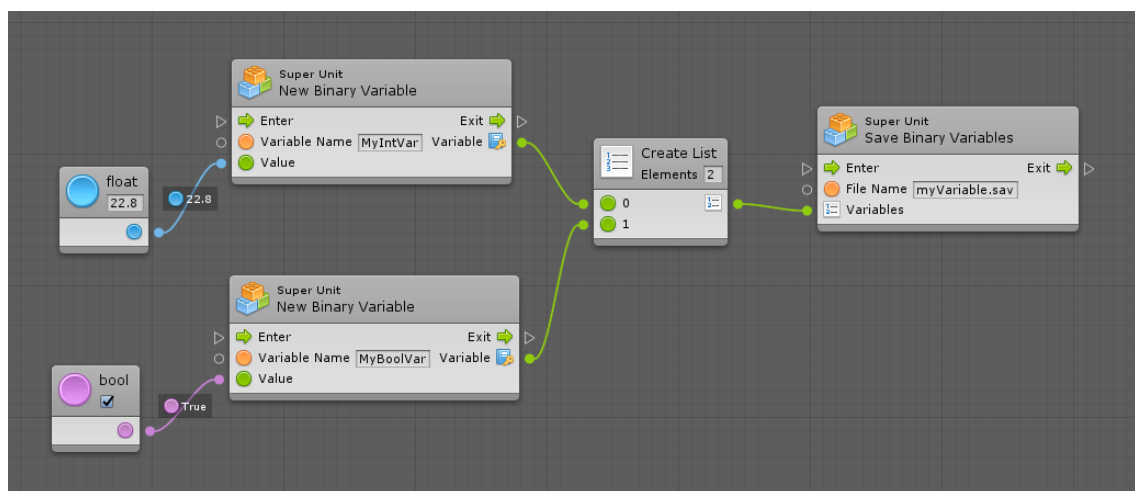


Figure 40: Visual representation of Load system

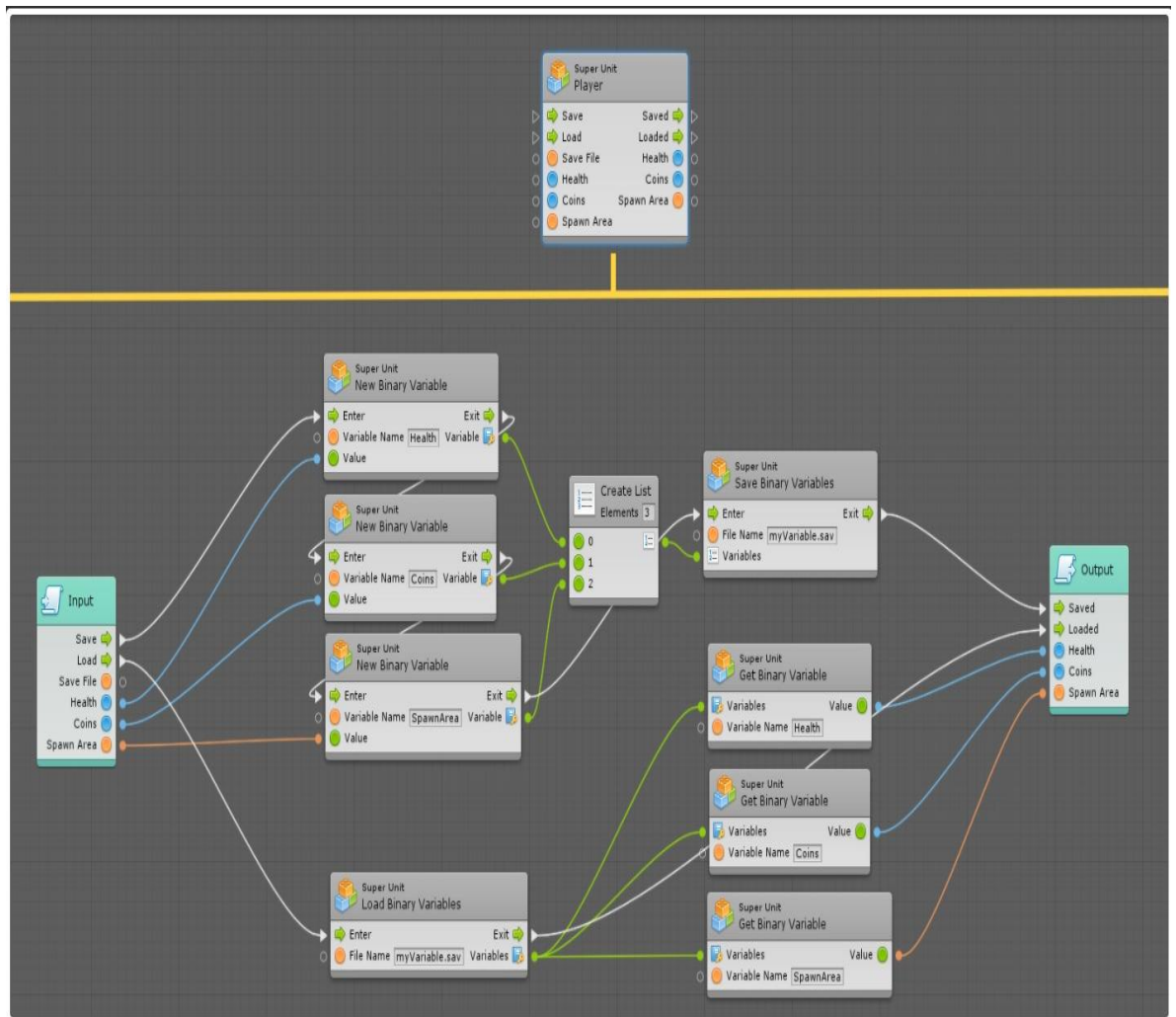


Figure 41: Visual representation of Save & Load system

5.2 Implementation of Front-end Design



Figure 42: Main Menu



Figure 43: Level Menu

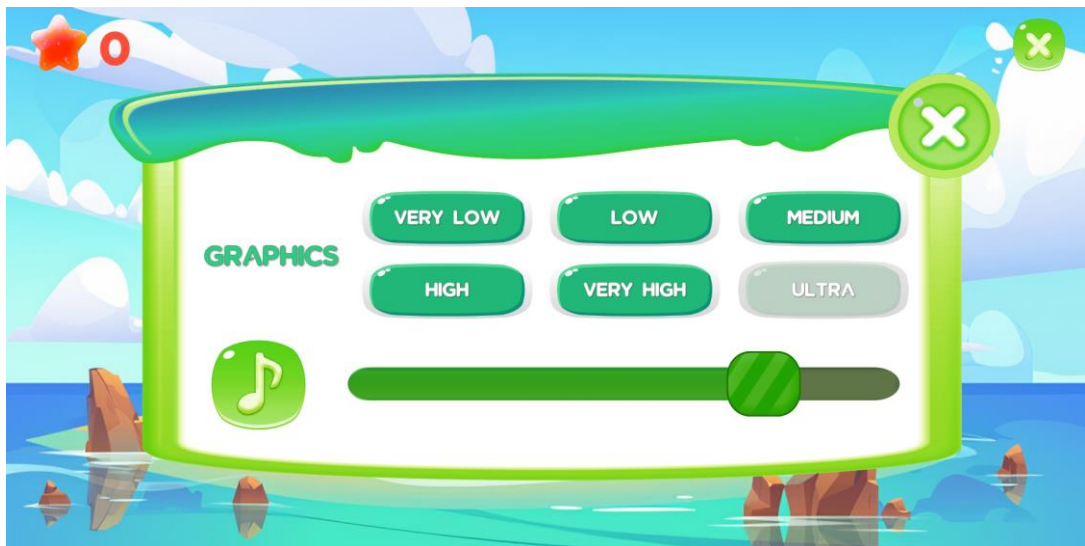


Figure 44: Settings Menu

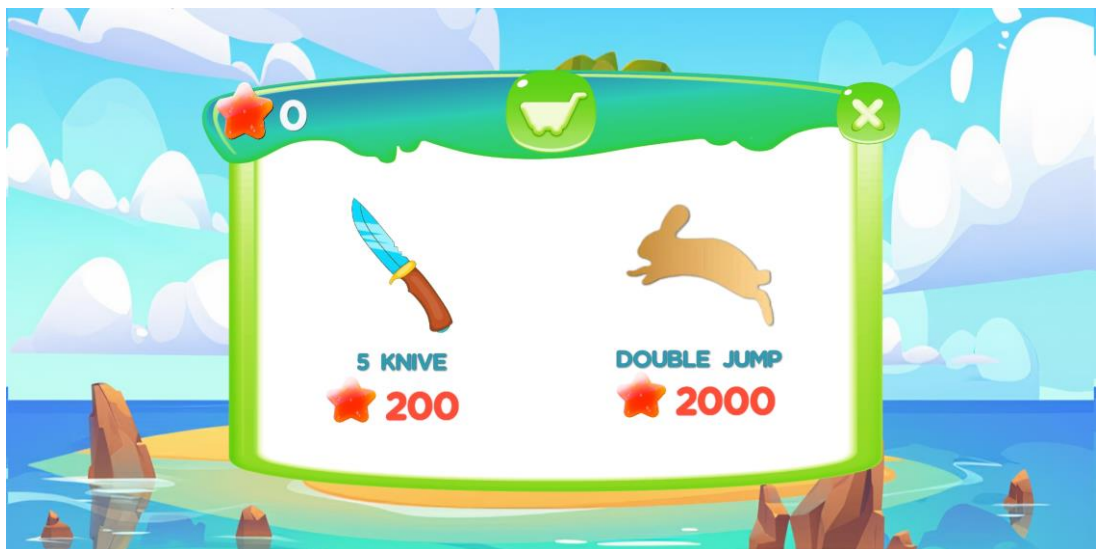


Figure 45: Store Menu



Figure 46: About Section

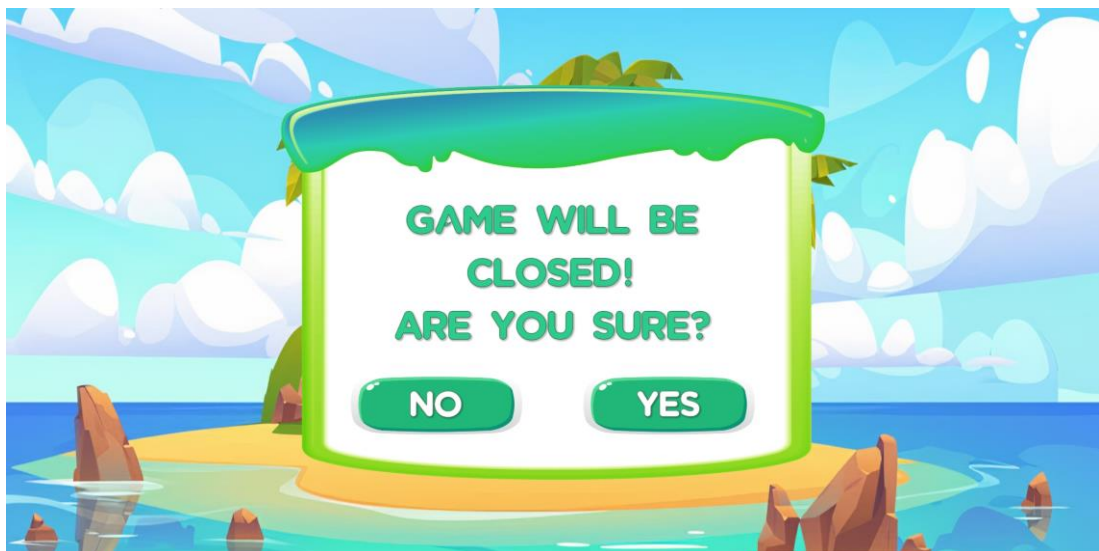


Figure 47: Warning Message

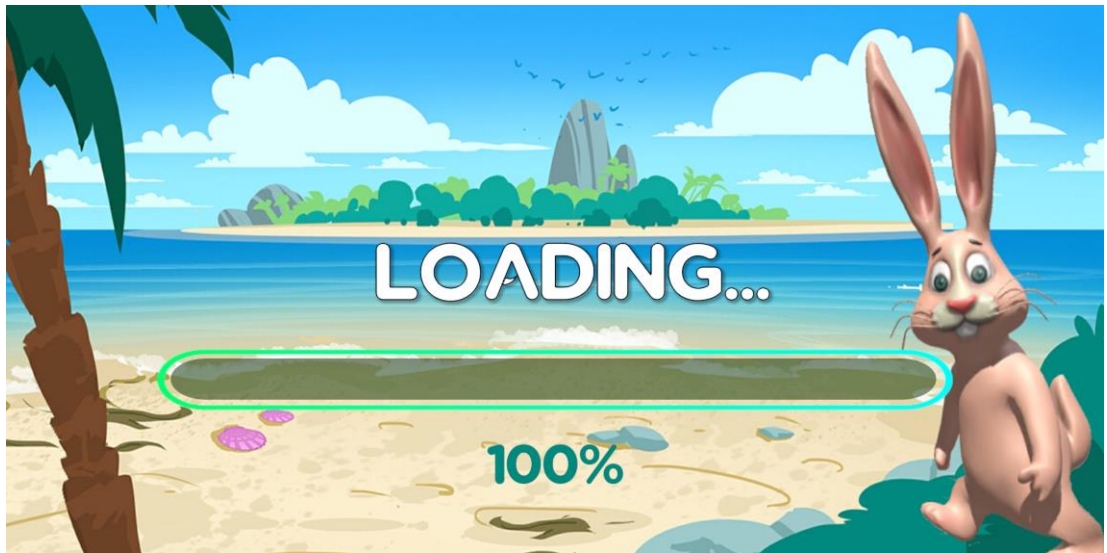


Figure 48: Loading Screen

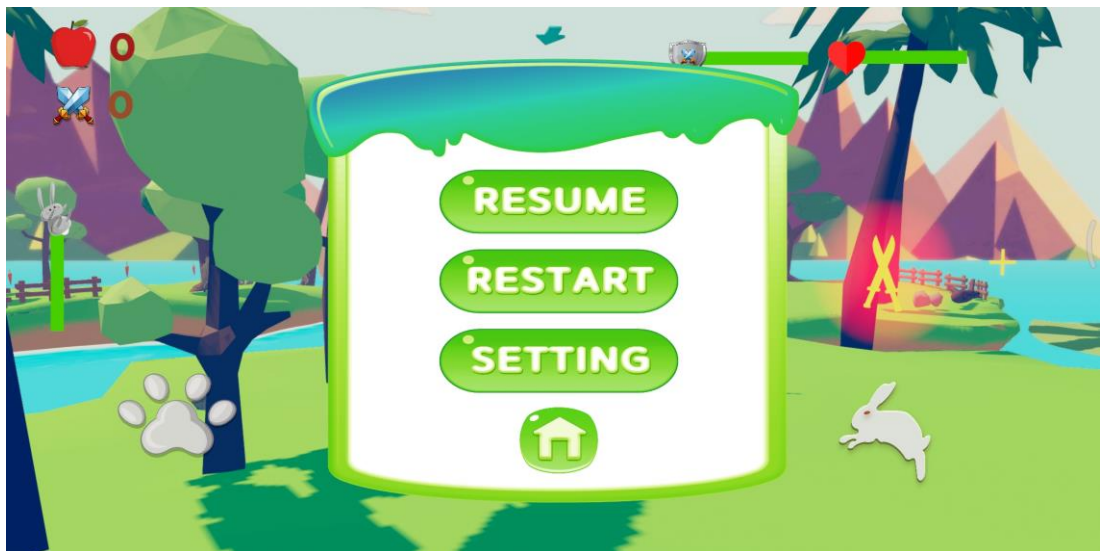


Figure 49: In-game Pause Menu

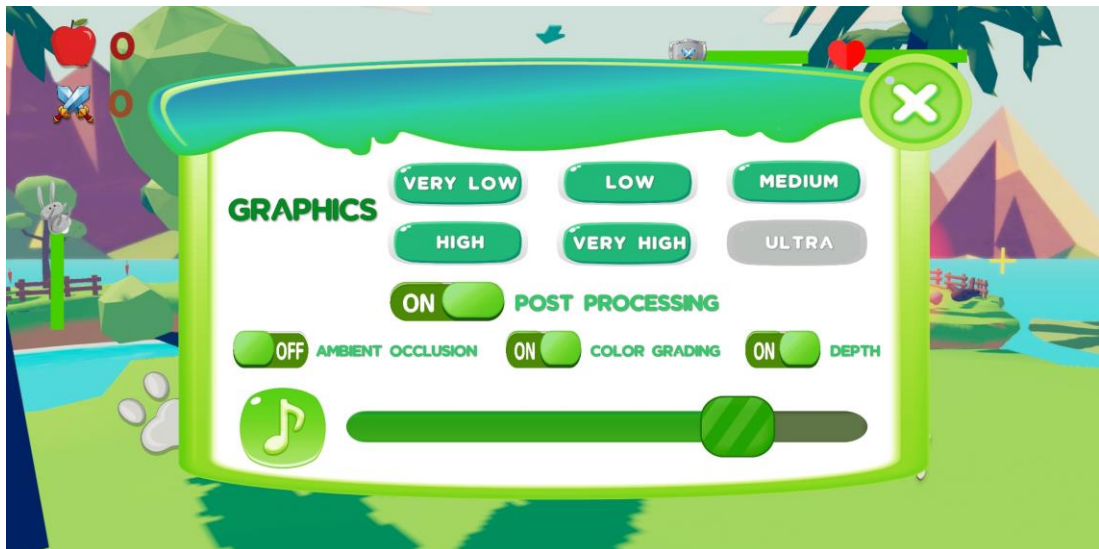


Figure 50: In-game Settings

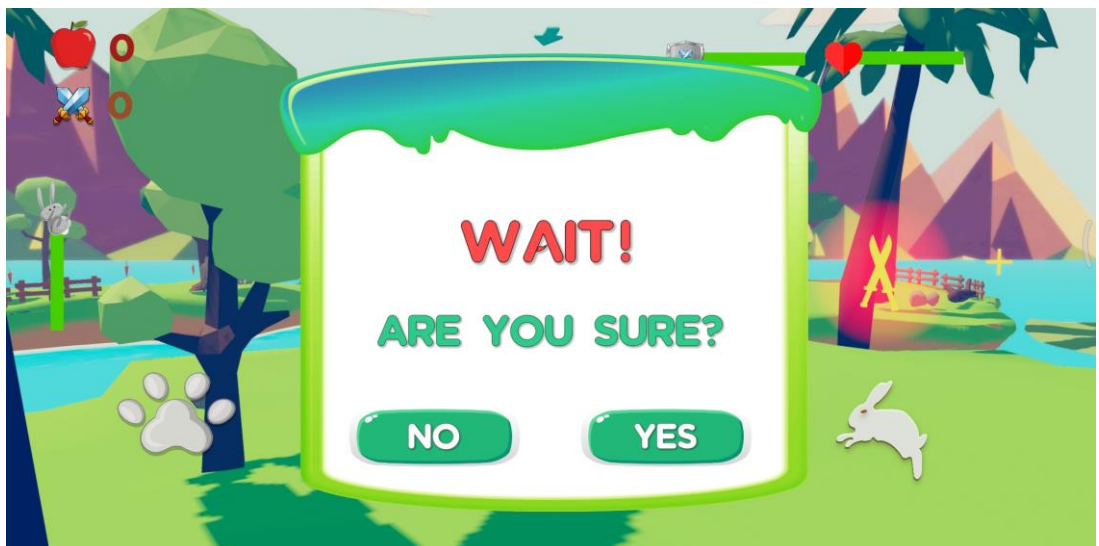


Figure 51: In-game Warning Message



Figure 52: Warning Message 2

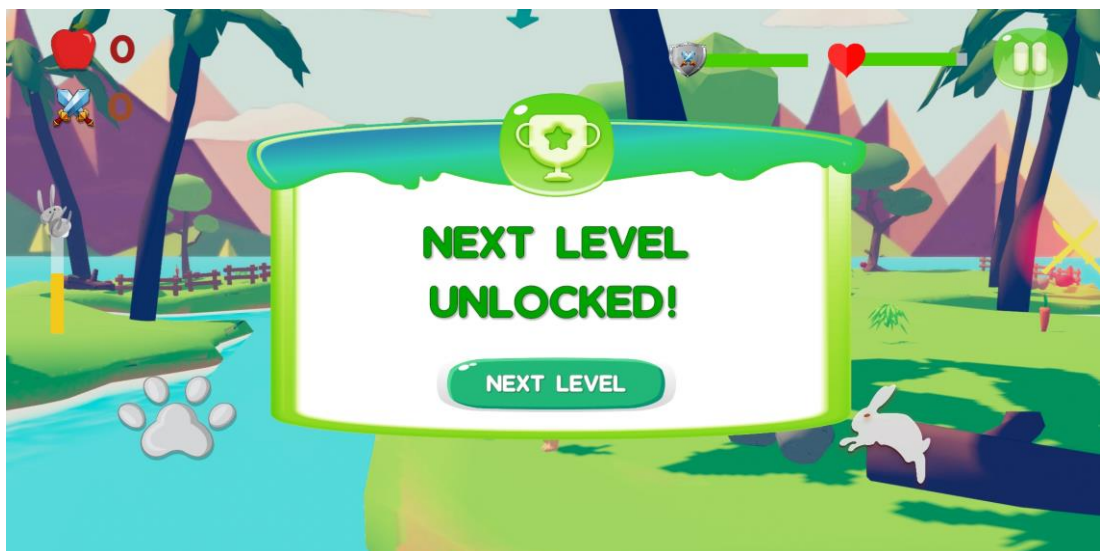


Figure 53: Level Finishing window

5.3 Testing Implementation

Game Testing is a software testing measure for testing computer or mobile games for quality control. The fundamental objective of Game testing is to recognize and find deformities and bugs in a computer game and work on the strength and execution. Game testing is a part of game development that assists with guaranteeing the computer game to be conveyed without bugs.

Testing games is a repetitive process and each new form might have bugs with and should be tried completely [8]. All game testing follows an essential design regardless of the size of the game and time needed for development. The game testers have to concentrate on game guidelines and prerequisites. Comprehend the general game part design and document engineering, the stream, and the record structures and the conditions related with the game. With each new model of the game, the test records should be regularly returned to refresh any adjustment of the details, new game testing experiments and new arrangement support [9]. A computer game tester needs to guarantee that no new issues were found. Testing implementation also checks the following,

Addressing the essential feature's availability and interest based on the planned storyline and target user group.

Recognizing the player and required framework and should be addressed into functional and non-functional requirements.

Recognizing Testable things, non-testable things and measures for practical and nonfunctional features.

Checking if the essential requirements are fulfilled,

Recognizing adjustable requirements and bugs.

Recognizing dependencies and requirements based on the uniqueness, difficulty and hardness.

Addressing the storyline of the game, gameplay, Characters, Animation, AI, Cinematic, Camera and special effects.

Types of game testing:

- i. Functional testing
- ii. Compatibility testing
- iii. Performance testing
- iv. Compliance testing
- v. Localization testing
- vi. Soak testing
- vii. Recovery testing
- viii. Security testing
- ix. Other game testing

5.3.1 Features that need to be tested

- i. Orientation checking (Landscape/Portrait).
- ii. Animation, movement, gesture checking.
- iii. Checking if there is any cropped background.
- iv. Checking object overlapping.
- v. Ensuring that splash screens are working.
- vi. Character's pathway limit should be checked.
- vii. Testing buttons and sliders.
- viii. Check for popup messages and labels.
- ix. Check for fonts and other visual elements.
- x. Performance
- xi. Scoring system
- xii. Multi Tasking
- xiii. Save Settings
- xiv. Data security

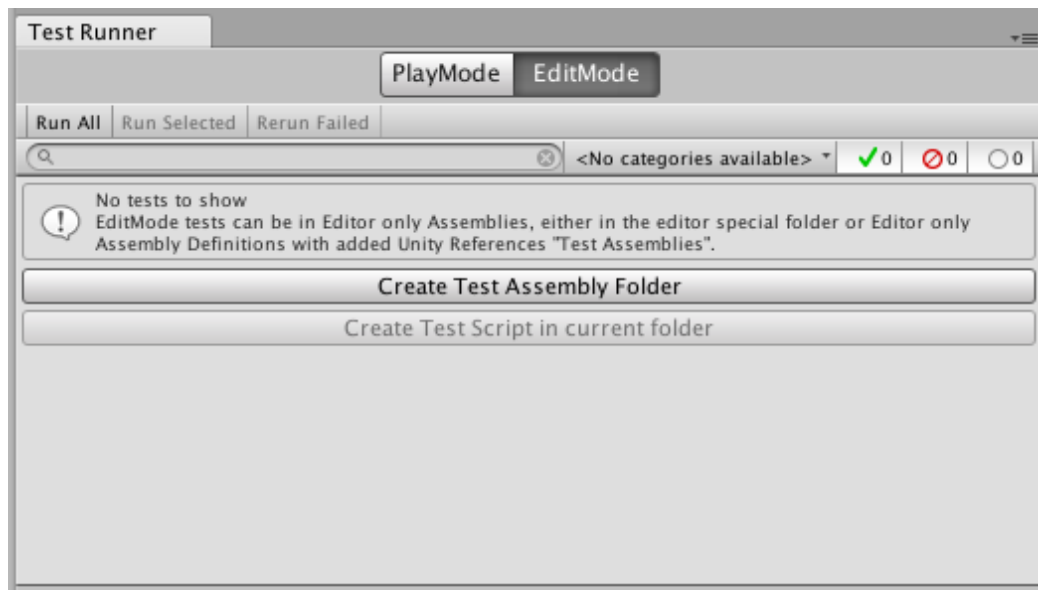


Figure 54: Unity Simulation Tool

5.4 Test Results and Reports

5.4.1 Simulation Test Result

Table 4: Test Case 1

Test case	This test checks if the player movement system is working correctly.
Test procedure	Moving the character in a loop through scripts in the scene.
Expected result	Perfect movement.
Test result	Everything was fine except jumping.
Comment	The jump distance should be decreased.
Conditional test	Run the scene again.
Expected result	This time it was working fine.
Test result	This time it was working fine.
Comment	Player stamina was affected.

Table 5: Test Case 2

Test case	Checking if the Colliders are working fine.
Test procedure	Arranging objects to the scene with colliders attached.
Expected result	Interactive colliding.
Test result	Everything worked fine.
Comment	In some cases, box colliders can be used to improve performance.
Conditional test	Not needed
Expected result	None
Test result	None
Comment	None

Table 6: Test Case 3

Test case	Checking if the animations are working.
Test procedure	Playing cutscenes and special scenes with animations.
Expected result	Proper animation play.
Test result	Everything was fine except the game over animation.
Comment	Player-Enemy collider was not working.
Conditional test	Run the scene again with little change to scene.
Expected result	Proper animation play.
Test result	This time it was working fine.
Comment	None

Table 7: Test Case 4

Test case	Checking if the weapon system is working.
Test procedure	Launching weapons at a regular interval through scripts.
Expected result	Precise target hit.
Test result	Everything was fine except the speed & velocity of the knife was low.
Comment	Knife throwing speed should be increased.
Conditional test	Run the scene again with little change to the script.

Expected result	Precise target hit.
Test result	This time it was working fine.
Comment	None

Table 8: Test Case 5

Test case	Checking if the Store is working.
Test procedure	Purchasing every item with a coin.
Expected result	Block purchase if insufficient coin.
Test result	Everything worked fine.
Comment	None
Conditional test	None
Expected result	None
Test result	None
Comment	None

Table 9: Test Case 6

Test case	Checking if the Music is working.
Test procedure	Playing every music through the SoundMixer.
Expected result	No overlapping track.
Test result	Everything was fine except the game over music.
Comment	Music name was misspelled.
Conditional test	Run the scene again with little change to the script.
Expected result	No overlapping track.
Test result	This time it was working fine.
Comment	Sound manager needed a check.

Table 10: Test Case 7

Test case	Checking if the graphics settings are working.
Test procedure	Toggling all graphics presets.
Expected result	Clearly noticeable visual differences.
Test result	Medium settings weren't rendering the shadows.

Comment	Medium and low preset were swapped.
Conditional test	Run the scene again with fixing the preset settings.
Expected result	Clearly noticeable visual differences.
Test result	This time it was working fine.
Comment	None

Table 11: Test Case 8

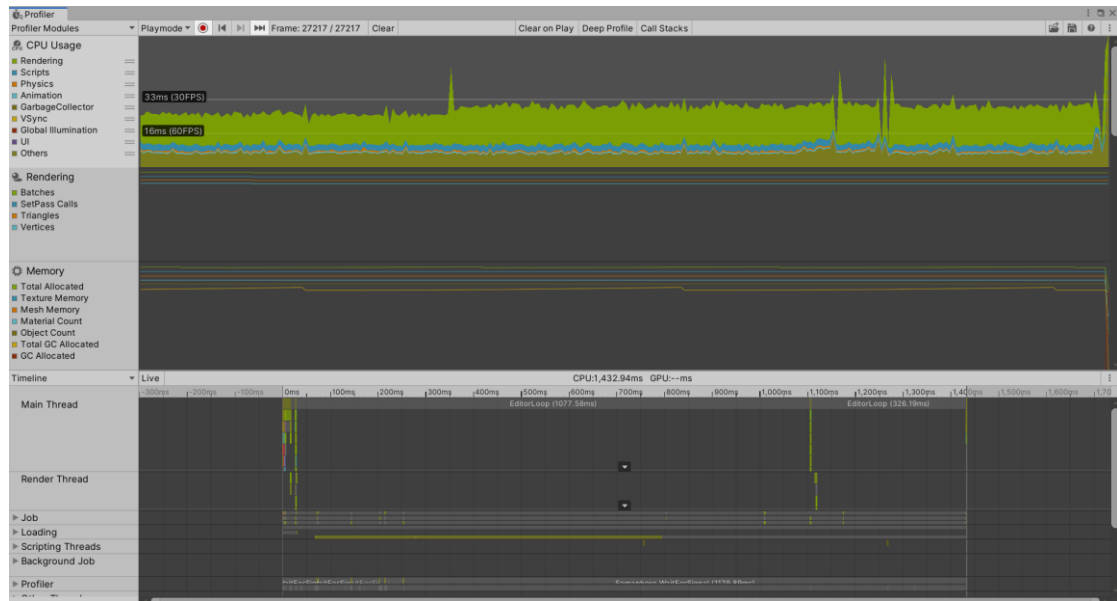
Test case	Checking if the UI is working.
Test procedure	Opening and every window and popup box.
Expected result	Interactive and responsive system.
Test result	Store menu alert was not working.
Comment	Scripts need to be checked.
Conditional test	Run the test again with changes to the script.
Expected result	Interactive and responsive system.
Test result	This time it was working fine.
Comment	None

Table 12: Test Case 9

Test case	Checking the performance.
Test procedure	Playing the game with a different configuration preset.
Expected result	Consistent fps.
Test result	Post processing effect is not working in some preset.
Comment	Post processing volume needs to be checked.
Conditional test	Run the test again with changes made.
Expected result	Consistent fps.
Test result	This time it was working fine.
Comment	None

Table 13: Test Case 10

Test case	Checking the Post processing effects.
Test procedure	Checking Ambient occlusion, depth of field and color grading.
Expected result	Perfect visual on every configuration preset.
Test result	Post processing effect is not working in some low-config preset.
Comment	Post processing volume needs to be checked.
Conditional test	Run the test again with changes made.
Expected result	Perfect visual on every configuration preset.
Test result	Working fine.
Comment	Post processing should be turned off on low configuration devices.

**Figure 55: Unity Profiler for testing**

Apart from Unity’s simulation testing, the game is also played and tested by some real users. To match the genre of the game, the users were specially picked from the age group of 10-15. Every one of them played at least 50-70% of the total missions on different smartphones. After that, their feedback was well noted.

5.4.2 Real User Review

5.4.2.1 User 1 review:

Age: 12

Gender: Male

Device: Xiaomi Redmi Note 9

Total progress: 80%

Feedback on Controlling: Okay

Feedback on Gameplay: Could be easier

Feedback on Characters: Perfect

Feedback on Story: Good

Feedback on Graphics: Good

Feedback on Music: Okay

Feedback on Mission Complexity: Okay

Feedback on UI: Good

Comment: Gameplay would be easier

5.4.2.2 User 2 review:

Age: 15

Gender: Female

Device: Samsung M30

Total progress: 60%

Feedback on Controlling: Good

Feedback on Gameplay: Okay

Feedback on Characters: Perfect
Feedback on Story: Good
Feedback on Graphics: Good
Feedback on Music: Good
Feedback on Mission Complexity: Okay
Comment: Liked the overall gaming experience

5.4.2.3 User 3 review:

Age: 14
Gender: Male
Device: Poco F1
Total progress: 90%
Feedback on Controlling: Okay
Feedback on Gameplay: Okay
Feedback on Characters: Perfect
Feedback on Story: Good
Feedback on Graphics: Okay
Feedback on Music: Good
Feedback on Mission Complexity: Okay
Feedback on UI: Okay
Comment: None

5.4.2.4 User 4 review:

Age: 11
Gender: Male
Device: Redmi Note 7 pro
Total progress: 50%

Feedback on Controlling: Okay
Feedback on Gameplay: Good
Feedback on Characters: Perfect
Feedback on Story: Good
Feedback on Graphics: Okay
Feedback on Music: Fine
Feedback on Mission Complexity: Okay
Feedback on UI: Good
Comment: None

5.4.2.5 User 5 review:

Age: 12
Gender: Female
Device: Samsung Galaxy S9
Total progress: 60%
Feedback on Controlling: Good
Feedback on Gameplay: Good
Feedback on Characters: Perfect
Feedback on Story: Good
Feedback on Graphics: Okay
Feedback on Music: Good
Feedback on Mission Complexity: Okay
Feedback on UI: Good
Comment: Needs more mission

5.4.2.6 User 6 review:

Age: 16

Gender: Female

Device: Realme 6 Pro

Total progress: 40%

Feedback on Controlling: Good

Feedback on Gameplay: Good

Feedback on Characters: Good

Feedback on Story: Good

Feedback on Graphics: Good

Feedback on Music: Okay

Feedback on Mission Complexity: Okay

Feedback on UI: Good

Comment: None

5.4.2.7 User 7 review:

Age: 18

Gender: Male

Device: Xiaomi K20 Pro

Total progress: 70%

Feedback on Controlling: Good

Feedback on Gameplay: okay

Feedback on Characters: Good

Feedback on Story: Good

Feedback on Graphics: Good

Feedback on Music: Okay

Feedback on Mission Complexity: Okay

Feedback on UI: Okay

Comment: Needs more mission

5.4.2.8 User 8 review:

Age: 15

Gender: Female

Device: Redmi Note 8

Total progress: 40%

Feedback on Controlling: Good

Feedback on Gameplay: Good

Feedback on Characters: Good

Feedback on Story: Good

Feedback on Graphics: Good

Feedback on Music: Okay

Feedback on Mission Complexity: Okay

Feedback on UI: Okay

Comment: Needs more music

5.4.2.9 User 9 review:

Age: 13

Gender: Male

Device: Redmi Note 5

Total progress: 40%

Feedback on Controlling: Good

Feedback on Gameplay: Okay

Feedback on Characters: Good

Feedback on Story: Good

Feedback on Graphics: Okay

Feedback on Music: Okay

Feedback on Mission Complexity: Okay

Feedback on UI: Okay

Comment: Needs multiplayer features

5.4.2.10 User 10 review:

Age: 15

Gender: Male

Device: Redmi Note 9 pro

Total progress: 70%

Feedback on Controlling: Good

Feedback on Gameplay: Good

Feedback on Characters: Good

Feedback on Story: Okay

Feedback on Graphics: Good

Feedback on Music: Good

Feedback on Mission Complexity: Fine

Feedback on UI: Okay

Comment: None

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on society

Over the years, video gamers have been a huge and increasing part of our way of life.

There are many advantages that gaming can have on society overall. In an internet based survey, gamers were asked what they think to be the primary advantage of gaming. More than 40% said that gaming can improve their mental wellbeing. This is a big difference to a very common view that playing computer games prompts violence and rude behaviour. Which, as per Qutee, is a view that 93% of gamers believe is mistaken. Qutee is an organization which conducted an online poll about this issue. It is a data-led discussion platform that is giving the gamers a voice and asking them for their opinions on the influence that games have had on their lives. One advantage that was discussed was making solid friendships. Around 66% of gamers guarantee that they have gotten together with five friends while playing and around 33% say they have made considerably more [3]. Playing a game gives a decent icebreaker, shared view, and an approach to get to know one another. It's time we began paying attention to individuals that really play these games as opposed to individuals that simply need somebody to blame for the antisocial behavior they see on the news. Ofcourse, gaming is anything but an ideal world, there are flaws, for example, gaming addiction that should be controlled. However, generally, it appears to be that gaming has a positive effect and should be treated positively.

Following are a few important factors that are related to gaming:

- i. Community build up
- ii. Stress release
- iii. Skill development
- iv. Finding friends
- v. Hobby
- vi. Enjoyment
- vii. Problem solving

This particular game is specially made for kids. The game contains some exciting and challenging missions with different levels of difficulties. By playing this game, they will get better at taking on new challenges and keep patience. Also to complete the game levels, they need to use the ingame navigation system which will only make them better at using navigation in real life. In that sense, this game is somewhat educational also. So the impact of this game is only going to be positive for society.

The Benefits of Gaming

Beneficial to Society



89%
believe gaming
is beneficial
to society



44%
say the most important
benefit is improved
emotional well being



93%
say the media's obsession
with linking violence and
video games is not justified

Making Friends



66%
say they made
up to 5 friends
while gaming



37%
say they made
more than 5
friends

Other Benefits



30%
say gaming
improves
cognitive skills



33%
say it inspired future
careers including
history, art and science



Forbes

Figure 56: Impact & benefits of gaming in society [Forbes]

6.2 Impact on Environment

From plastic packaging to electricity consumption, computer games can sometimes badly affect the environment. This stems to a great extent from the power that playing devices require, which so regularly come from non-renewable energy source power plants which radiate carbon dioxide. While taking into account what gaming means for the climate, a few variables become possibly the most important factor, including the actual misuse of packaging, the forced production of various gaming consoles, and surprisingly the length of various computer games when it impacts time spent playing.

Computer game makers are turning out to be progressively mindful of their current wasteful state and the significance of decreasing it. For example, Ubisoft stopped actual guidance manuals in 2010, and Microsoft's obligation to carbon reduction began in 2012. Both physical and computerized games can leave a shockingly carbon impression, as per Console Carbon Footprint research led by Slots Online Canada. For actual games, the most noticeably terrible guilty party is the bundling, including the aluminum poly-carbonate plate, the plastic case, and the paper cover and pamphlet inside the case. These can wind up as rubbish, particularly on account of yearly games like EA Sports series. As indicated by the Console Carbon Footprint research, if all FIFA 19 players discarded their plates when buying FIFA 20, they would produce more than 283,000 pounds of landfill waste and rubbish [1]. In the meantime, advanced games can require extra energy utilization dependent on download times and server maintenance. Although computerized downloads are turning out to be more normal and are more eco-accommodating than actual games, that's not enough. After being introduced by means of physical disc or download, a game's effect on the climate is generally a consequence of energy utilization. As such, the additional time players go through with their playing devices turned on and connected, the more energy is burned-through from power establishments that transmit CO₂ [2]. Likewise, the increasing popularity and demand of cloud gaming puts additional weight on remote servers. With this web based streaming, servers often handle the graphics processing remotely as well to reduce carbon footprint.

When it comes to this particular game, players do not need any extra console or any other devices as it is a mobile game and smartphones are very common these days. Also the game is only available on web servers so no physical packaging is needed which is better for the environment. Also mobile devices consume less power which will not be electricity hungry thus having almost no negative impact on fossil fuel energy consumption.

6.3 Ethical Aspects

One of the most discussed subjects in the game industry is not the graphics, storytelling or ongoing gameplay, but rather ethics. People, no matter if they are gamers or not, often wonder whether the game developers have a moral obligation while making their next game. Actually just like the individuals behind movies or TV shows, game developers are approached to consider how their work will impact and influence people.

Often, it appears to be that the general opinion likes to focus on the morals of computer games considerably more than that of movies and TV shows, maybe because it's a newer and more up to date medium of entertainment [10]. Computer games permit the player to simply decide and basically become a person as they want which is not even possible in movies or TV shows. Many argue that the independent, immersive, and experiential nature of this type of amusement raises extraordinary moral issues.

Apparently, the game industry sees games not only as a medium of amusement, but also as an artistic expression and that's why in many cases, there were incidents of misuse of ethics.

Here are some ethical aspects regarding video games:

- i. Video game addiction
- ii. Looting and gambling
- iii. Violence

- iv. Dark Storytelling
- v. Explicit visuals

This particular game is expected to be played by kids. So the story is simple and subtle while having zero explicit content, rather animated and cartoon characters and visuals. The game story and gameplay has a funny and enjoyable type of attacking and self defence system which is not violent. The missions and levels are short and precise so that it won't take hours to complete thus reducing the possibility of addiction of playing this game.

6.4 Sustainability Plan

A good game needs good planning to sustain.

Emphasizing on story

Often game developers make such games that have a poor storyline. They don't think about substantial issues that players face during development. This is one of the biggest reasons behind a game being unsuccessful.

That is why the story of this game is designed in a way that it will always attract the player to the next mission.

Establishing points of difference

People are continually searching for a new thing. They tend to get bored of playing games that tell the same storyline.

Also, in terms of mobile games, there are now a lot of games accessible on the PlayStore. Players need to be presented with something which is exceptional.

Accordingly, we need to make something interesting that draws the player's attention and keeps them engaged. That's why it's a 3d adventure game instead of being a normal 2d one.

Design scalability

Different blunders happen at whatever point the designers change the interface depending on client expectations. Additionally, make adaptable plans so that any component can be added any time during the versatile application advancement measure. In this case, the design is dynamic and fully compatible with any screen size to avoid future complications.

Frequent updates

Updating a game frequently is vital. To assemble experiences about the client point of view, input feedback has a crucial part.

While refreshing all the applications, it's really important to add the latest and greatest functionalities with the main target of further developing the player's experience. The improvement group should understand its significance.

Focusing on user playback

At the appropriate time, developers can contribute to the client's criticism request to help their presumption and change it further. It is essential to listen to what users want and how they feel about the game and improve accordingly.

Working on efficiency and performance

A video game's user experience is massively dependent on its performance and efficiency. The game has to be optimised and only get better with time. This is important to keep the players attached to the game.

Becoming a master of one platform

A single game can be distributed to different platforms like Android, IOS, PC, Playstation or Nintendo but it has some problems. The broader section of platforms and its users may create some issues like compatibility and performance. Also it is essential to keep the game relevant for a longer period of time. That's why it's only getting released on Playstore for Android devices.

Promotions and advertisements are also vital for an Android game to be successful these days. It will be done mainly on social media platforms to get a better reach.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion

‘Bunny’s Fortune’ is a game that is somewhat unique if we only consider the mobile platform. There are a lot of mobile games that have the features of a basic platformer game but most of them are made in 2d or 2.5d world space. Some popular titles are available with 3d world space but most of them have the gameplay of an endless runner game. The fantasy-adventure genre is yet to be used in 3d mobile games on a big scale. There are many similar 3d strategy games available but on the PC or console platform. This game is bringing that sort of less used genre to the mobile platform. Also it is a free to play game so anyone can play it anytime with ease. That gives the high hope of the game being hit. But there is still a lot to improve. This is the first release, so it may have some issues and bugs here and there. Players may feel the need for some change in controlling and gameplay. Though the first release is already tested by some players and has some bugs already fixed, it still needs more feedback and probably more adjustments made accordingly to get better.

7.1.1 The Obstacles

- i. Working with a Game Engine is a completely new experience as they have a lot of features, tools and modules. Normally as programmers, we work with different object oriented languages, web development and database

management systems but cooing up with a Game Engine was a completely new thing and bit tough to catch up.

- ii. Creating and rigging a 3d model is not an easy task as it takes a lot of time to work with each and every vertice of the model.
- iii. As it is a new field, a lot of learning and practice was needed to adapt every single new tool.
- iv. Almost every single bug and glitch took a long time to fix as it took a lot of trying, playtesting and script changing every time.
- v. As a programmer, working with animation systems and linking them up through lines of code was not so easy to implement.

7.1.2 The Achievements

- i. Working with a full scale Game Engine and the experiences and knowledge gained from it is useful for future projects.
- ii. Creating full scale games from scratch by maintaining and following the standard software development life cycle is something to be proud of.
- iii. Improvement of programming skills with C#.
- iv. Improvement of problem solving capability.
- v. Improvement of thinking and imagining ability.

As we can see, mobile games are very popular nowadays, which makes developers more and more interested in making new games. So the number of games is only increasing day by day and it makes the success of new games uncertain. But at the same time, the number of mobile gamers is also increasing. So it all gives hope. Even if it performs unexpectedly in the market, the knowledge and experiences gathered from the development of this project will always be really useful. Also is it something different from the usual games available from our local developers. Most importantly, playing a self made full scale game with family and friends is something very enjoyable and always inspiring to do even better in the next projects.

7.2 Scope for Further Developments

Improving graphics: Video games are all about graphics these days. No matter whether it's a 2d or 3d game, the graphics, art and visuals are vital for a game to be successful. This game is based on a three dimensional environment so the scope for further graphics improvement is huge. Improving graphics can include more vivid and detailed scenes to the game, better character models and items.

Improving animation: Animation is important and it makes the scenes more intense and interesting. So far the game has a good number of animation clips here and there but it can be always increased.

Improving UI: The game has a fixed pattern of design. Every UI element has a particular look and theme to them. But it can be more dynamic and interactive which improves the user experiences.

Adding multiplayer functionality: Nowadays, multiplayer features are very important and popular in mobile games. It can also be introduced to this game. For that, some adjustments in the storyline and gameplay will be needed but it can be achieved through few updates.

Adding leaderboard: It's a popular practice in any game. Player's progress can be listed in a real time ranking which makes the gaming experience more exciting. It can be added to this game with future updates.

Social media interaction: Adding the option to link the game with social media will be added to this game soon. It not only makes the game easier to backup and restore when needed but also does the job of promotion.

All together, there are many scope of further improvements and development to make the game better with time.

REFERENCE

- [1] A. HAGOOD, "Playing Video Games Can Be Bad For The Environment (Yes, Really)," Screenrant, 25 February 2020. [Online]. Available: <https://screenrant.com/video-game-environmental-impact-reports-carbon-footprint-energy/>. [Accessed 29 September 2021].
- [2] L. Gordon, "THE MANY WAYS VIDEO GAME DEVELOPMENT IMPACTS THE CLIMATE CRISIS," The Verge, 5 May 2020. [Online]. Available: <https://www.theverge.com/2020/5/5/21243285/video-games-climate-crisis-impact-xbox-playstation-developers>. [Accessed 2 October 2020].
- [3] K. Anderton, "The Impact Of Gaming: A Benefit To Society," 25 June 2018. [Online]. Available: <https://www.forbes.com/sites/kevinanderton/2018/06/25/the-impact-of-gaming-a-benefit-to-society-infographic/?sh=1a1d2114269d>. [Accessed 30 September 2021].
- [4] D. Pickell, "The 7 Stages of Game Development," g2.com, 15 October 2019. [Online]. Available: <https://www.g2.com/articles/stages-of-game-development>. [Accessed 1 October 2021].
- [5] D. Sekulic, "Efficient Occlusion Culling," NVIDIA, [Online]. Available: <https://developer.nvidia.com/gpugems/gpugems/part-v-performance-and-practicalities/chapter-29-efficient-occlusion-culling>. [Accessed 2 October 2021].
- [6] R. T. Bonet, "Level of Detail (LOD): Quick Tutorial," Game Developer, 18 June 2021. [Online]. Available: <https://www.gamedeveloper.com/disciplines/level-of-detail-lod-quick-tutorial>. [Accessed 28 September 2021].

- [7] H. Shaffer, "What's the Deal with Low Poly Art?," NOD, 15 January 2015. [Online]. Available: <https://www.sessions.edu/notes-on-design/whats-the-deal-with-low-poly-art/>. [Accessed 29 September 2015].
- [8] Test Bytes, "9 Different Types of Game Testing Techniques," Test Bytes, 6 July 2017. [Online]. Available: <https://www.testbytes.net/blog/types-of-game-testing/>. [Accessed 3 October 2021].
- [9] N. Limbachiya, "7 Different Types of Game Testing Techniques," DZone, 13 August 2020. [Online]. Available: <https://dzone.com/articles/7-different-types-of-game-testing-techniques>. [Accessed 2 October 2021].
- [10] C. Hodent, "Ethics in the Videogame Industry: A Mythbusting and Scientific Approach," Celia Hodent, 18 December 2019. [Online]. Available: <https://celiahodent.com/ethics-in-the-videogame-industry/>. [Accessed 29 September 2021].