

MilkyWay



Developer(s): Abdul Rauf (BO460872)

Muhammad Nouman Shahid
(BO461116)

Muhammad Asif Moeen (BO461115)

Supervisor(s): Mr. Furqan Ahmed Ghauri

{ Bahawalpur }

ALLAMA IQBAL OPEN UNIVERSITY

H-8 ISLAMABAD

Year 2021

Declaration

We hereby declare that this software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we developed this software, and this report entirely on the basis of our effort made under the sincere guidance of our project supervisor.

If any part of this software proved to be copied or found to be a report of some other, we shall stand by the consequences.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other University or institute of learning.

We further declare that this software and all associated documents, reports, and records are submitted as partial requirements for the degree of BS (CS).

We understand and transfer copyrights for these materials to Allama Iqbal Open University, Islamabad.

Abdul Rauf (BO460872)

Signature: _____

Muhammad Nouman Shahid (BO461116)

Signature: _____

Muhammad Asif Moeen (BO461115)

Signature: _____

Final Approval

This is to certify that we have read the project report submitted by **Abdul Rauf, Muhammad Nouman Shahid, Muhammad Asif Moeen**, and in our judgment that this report is of sufficient standard to warrant its acceptance by Allama Iqbal Open University, Islamabad, for BS (CS) degree.

Examination Committee

Internal Supervisor

Furqan Ahmed Ghauri

AIOU, Islamabad

=====

Internal Examiner

=====

AIOU, Islamabad

=====

External Examiner

=====

AIOU, Islamabad

=====

ACKNOWLEDGMENT

In the name of ALLAH Almighty, most Gracious, most Merciful who has blessed us with the physical, and mental capabilities to complete the assigned project.

We would like to thank our respected supervisor, Mr. Furqan Ahmed Ghauri, whose guidance help and precious time helped us a lot in completing the given task.

He was the person who was always there with the prudent of advice, and guidance whenever we were in the need of it.

We must say his kind supervision appeared as the wind beneath wings, which kept us up, and upon every step.

Finally our heartiest gratitude to our kind and beloved parents who provided us with a chance to be in a position to do this project work and, who supported us through the ups, and downs in the duration of the project

Abdul Rauf (BO460872)

=====

Muhammad Nouman Shahid (BO461116)

=====

Muhammad Asif Moeen (BO461115)

=====

Project in Brief

Project Title:	MilkyWay
Organization:	Game Development
Objectives:	To keep children interactive with mobile forcing them to stay home.
Developer(s):	<ol style="list-style-type: none">1. Abdul Rauf2. Muhammad Nouman Shahid3. Muhammad Asif Moeen
Supervisor(s):	Mr. Furqan Ahmed Ghauri
Tools Used:	Godot game engine GD Script GD Shaders Krita Blender
Operating Systems:	● Android
System Used:	● Development: <ol style="list-style-type: none">1. Intel Xeon W3680 6-core 12-threads 3.33GHz 12MB L3 cache.2. 12GB RAM3. GTX-960 2GB ASUS OC CU-II Edition ● Target: <ol style="list-style-type: none">1. Android KitKat2. cortex A73. Mali 400

Abstract

“MilkyWay” is an Android video game developed to keep children entertained, and interacted at home so they do not leave home to play.

The Game has five levels, two-player’s ships, one fire type, two fire power-ups, a life system, four enemies including one boss and two modes i.e. endless and story, and a proper difficulty system.

The Game is aimed to keep the children interacted for about 30 minutes (half-hour).

Table of Contents

1	Introduction	8
1.1	Problem Definition	8
1.2	Scope of the Project	8
1.3	Broad Based Objectives	9
1.4	Specific Objectives	9
1.5	Feasibility Report	10
2	Existing System Analysis	11
2.1	Detailed Study of Existing System	11
2.2	Drawbacks in Existing System	11
2.3	Main analysis Diagram	12
3	Proposed System	13
3.1	Objectives	13
3.2	System Proposal	15
3.3	Benefits of Proposed System	16
3.4	Main System Diagram	16
4	System Design	17
4.1	Input Design	17
4.2	Output Design	18
4.3	Code Design	19
4.4	User Interface Design	21
5	Game Development	23
5.1	Customer requirements	23
5.2	Development Method	24
5.3	Planning	25
5.4	Publishing	34
6	System Testing and Evaluation	35
6.1	Introduction	35
6.2	Code Testing	35
6.3	Project Testing	39
6.4	Project Reviews	40
7	Future Extension	41
8	User Guide	43
9	References	49
10	Appendices	50

1-Introduction

Outline:

- 1 Problem Definition**
- 2 Scope of the Project**
- 3 Broad Based Objectives**
- 4 Specific Objectives**
- 5 Feasibility Report**
 - 5.1 Economic Feasibility**
 - 5.2 Technical Feasibility**

1.1 Problem Definition

In rural and small cities, children go outside their homes to get entertained and enjoy or to play games. There are Video Gaming Centers at which you need token(s) to play the game many children go there as well. And many bad people also happen to come by at such places and in fact, many gaming center owners are close to bad people like drug addicts, only because they are their permanent customers. Children in such communities are welcomed openly with open arms but it is not good for them. Such a community gives birth to illegal desires in children's heart. Such a problem is very common with middle-class families.

Especially in COVID19, no one wants his/her child to go out so they can stay safe and keep social distancing. But the children have to be controlled not by force but by love.

1.2 Scope of the Projects

The project targets 3+ years of children as the audience and every parent with an Android mobile device running KitKat as minimum software version and 512 MB memory with a dual-core processor.

- In story mode it entertains children for about 15 to 30 minutes.
- Endless mode is endless so the user is interacted for as long as the player is not dead.
- Keeps children in sighted.

- Two Player's ships to choose from.
- Two fire types.

1.3 Broad Based Objectives

The Broader view of objectives is:

- To enter in Game Development Industry
- To keep children home so they do not get effected by COVID19.
- To provide an offline game which keeps children interacted.
- Provide optimized video games so they can be played on low-end Android devices.
- Main targeted audience is middle and poor class families children so they can breed their child themselves and bad environment effects as little as possible.

1.4 Specific Objectives

Specific Objectives are:

- Optimized for low-end devices
- easy to play game for 3 to 5 years.
- Provide a new tool to parents who cannot afford expensive computing devices for keeping their children in front of their eyes.

- Ads free product to enjoy for children as ads has never been intuitive especially for children. Because ads ruins the use experience and usability.

1.5 Feasibility Report

1.5.1 Economic Feasibility

The project is economically feasible. Neither the initial cost is considerable nor the maintenance. The Project can grow widely and an in-game store market can be added. The complete development process used open-source MIT or GPLv3 licensed tools which makes the project economically feasible as it indicates possibly the lowest cost. Though with the increase in project size workers will become expensive and so will the income increase as well.

1.5.2 Technical Feasibility

The Project is technically feasible because of the Spiral Win-Win process modal, it uses a modular approach and is highly maintainable.

2-Existing System Analysis

Outline:

- 1 Detailed Study of Existing System**
- 2 Drawbacks in Existing System**
- 3 Main Analysis Diagram**

2.1 Detailed Study of Existing System

Usually, a middle-class family's child gets 10-20 Pakistani Rupees each day and has several choices as, eat something, enjoy some swings, play video games at a gaming center or save money. The gaming machine usually has a MAME32, NeoGeo, or PlayStation 1 emulator on the computer. They were the systems of the 1990s that continue to this day and are still successful in ruining the children. Token coin expires at game over either you win or lose.

Children who go to video games also encourage others to go with them. Normally the game can only be played for half an hour and then the game is over.

The video game system is such that the shopkeeper has some coins called tokens. The child buys a coin from the shopkeeper and puts it in the video game machine and starts playing the game. Once children lose all the tokens they had they no longer can play the video game.

2.2 Drawbacks in Existing System

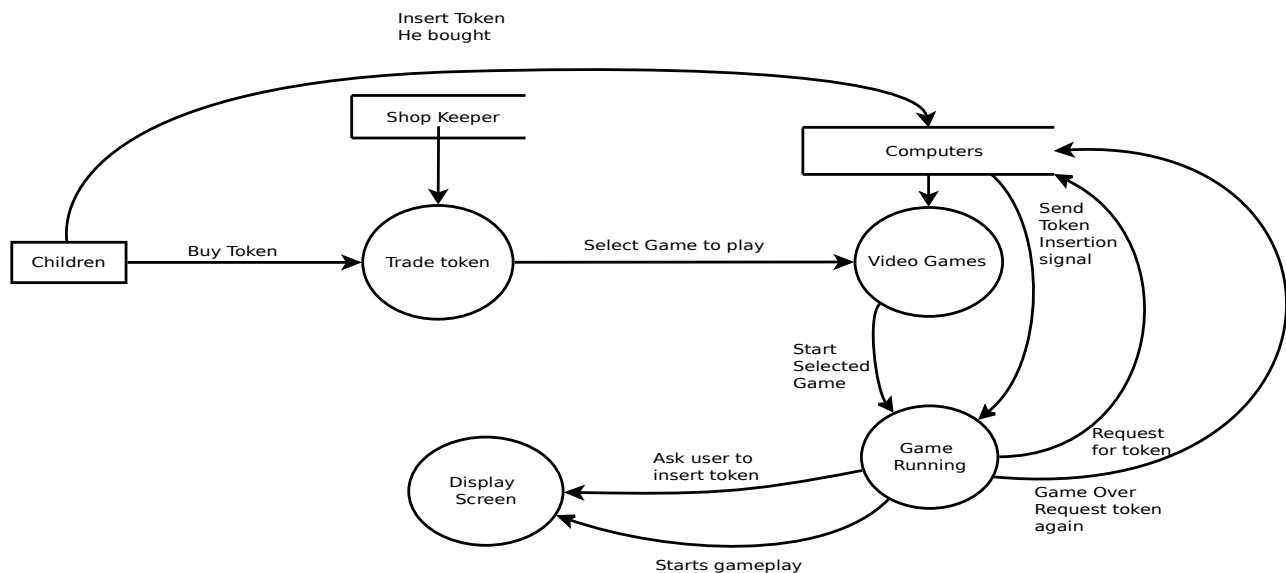
Existing system has several drawbacks some of them are as follows:

- Alcohol addicts also come to the game store.
- The child always gets into bad habits by going to the game store.
- Evil is always promoted in this place.

- Going there, the child always feels his poverty.
- The child has fun for 30 minutes and then watches other people for about 90 minutes.
- The child becomes distant from the parents.
- The child is always thinking of the game and no one else is doing it, it hurts the parents a lot.
- During this period the child disappears from the sight of the parents and causes trouble for all of them.

2.3 Main Analysis Diagram

Below is the DFD of current system. This explains how current system is working.



3-Proposed System

Outline:

1 Objectives

1.1 Encourage Children to stay home

1.2 Run-able on low-end Android devices

1.3 Ensure Children

1.4 Keeping children away from bad society

1.5 Unlimited Entertainment

1.6 Complete Offline experience

2 System Proposal

3 Benefits of proposed system

4 Main System Diagram

3.1 Objectives

Main objectives for the purposed system are listed as:

- Encourage Children to stay home
- Run-able on low-end android devices
- Ensure children interaction
- Keeping children away from bad society
- unlimited entertainment
- Complete offline experience

3.1.1 Encourage children to stay home

In this corona pandemic situation, it is not safe to let children go out because children are children who are careless and want to touch everything so this is important to make them stay home not forcefully but by providing them everything they need at home even entertainment.

3.1.2 Run-able on low-end Android devices

Allow middle class and poor parents who own low-end Android phones to get their child to entertain from their mobile so that child may not waste money on video games center.

3.1.3 Ensure children interaction

It is important to keep the child interacted for about 30 minutes at least to ensure the child's

entertainment. Because most parents let their children entertained for about 30 minutes after school and then again 30 minutes after tuition. So the child must spend all this duration on the game on a mobile phone so that he may not go out to enjoy it.

3.1.4 Keeping children away from bad society.

Alcohol addicts also tend to come to the game store and the presentation of children at such places among such people is very bad. It's not they come alcoholic but it's just that such people do happen to pass by such places.

3.1.5 Unlimited Entertainment

At game centers, children can only play for a limited time because it's pay to play strategy at those places. But I purpose a method which is overall much cheaper and also provides opportunities for unlimited entertainment.

3.1.6 Complete Offline experience

Internet is also not a good place for children because it shows adult ads everywhere throughout the internet and leads the child to adultery content. And poor parents also don't afford internet so a complete offline experience is what we propose.

3.2 System Proposal

We propose a video game named “MilkyWay” which is a space shooter game. It would have several features listed below:

- Two Game Modes
 - 1. Story
 - 2. Endless
- Two Player Ships
 - 1. Player 0
 - 2. Player 1
- Five Levels
- Four enemies
 - 1. virus
 - 2. todda
 - 3. fighter
 - 4. final boss
- Three difficulties
 - 1. Easy
 - 2. Medium
 - 3. Hard

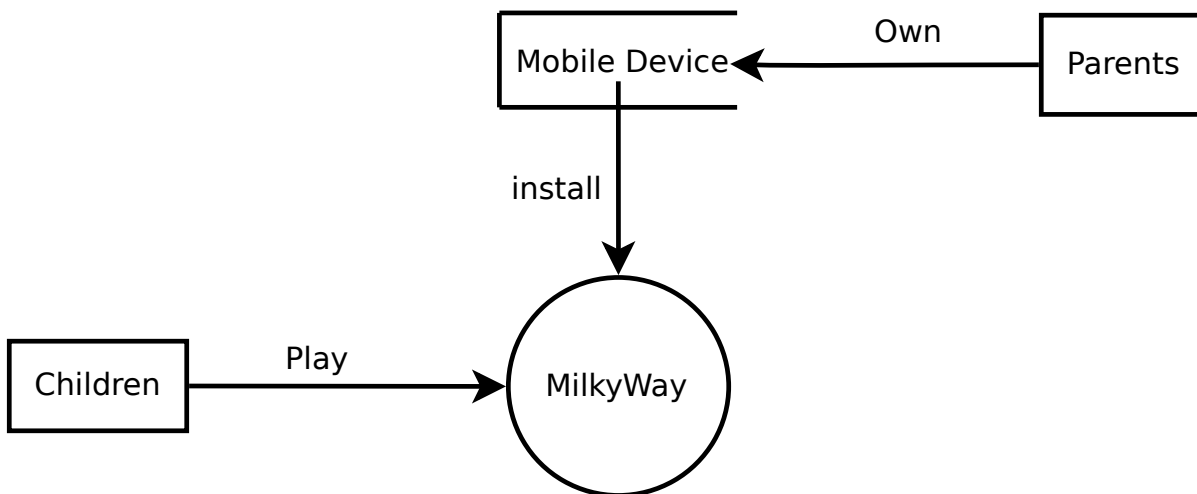
3.3 Benefits of Proposed System

Our proposed system has many benefits but several that instantly comes into mind are:

- Offline
- Executable on low-end Android devices
- Optimized for performance
- Small in size
- Developed for children
- Nice animations
- No Data collection
- Easy to play

3.4 Main System Diagram

This diagram shows the DFD of purposed system.



4-System Design

Outline:

- 1 Input Design**
- 2 Output Design**
- 3 Code Design**
 - 3.1 Object-Oriented approach**
 - 3.2 Integration of components**
 - 3.3 Design Pattern**
 - 3.4 Algorithms**
- 4 User Interface Design**
 - 4.1 Splash screen**
 - 4.2 Main Menu UI**
 - 4.3 Main Scene**

4.1 Input Design

Since MilkyWay is a video game at small scale, hence it does not have many inputs but whatever inputs are, they are listed as:

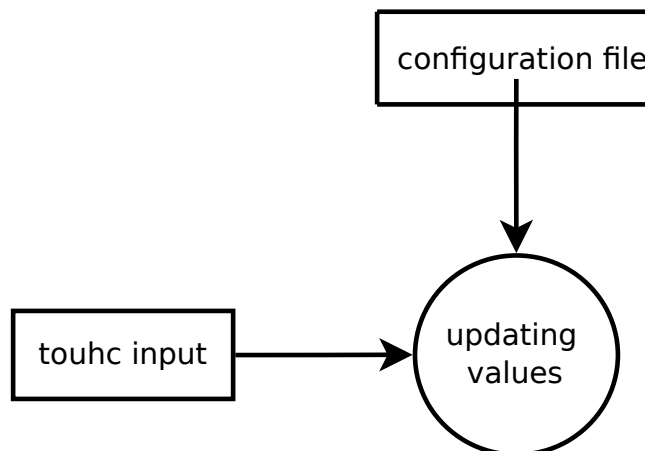
➤ configuration file

- ◆ the configuration file contains a high score since it's the only configuration being stored somewhere. Hence reading that file to update the high score.

➤ user touch input

- ◆ Since a video game highly depends on user input and android mobile devices can usually have touch inputs. So always keeping track of user input to avoid misbehavior.

➤ Below diagram explains inputs



4.2 Output Design

Unlike input game has several outputs listed as:

➤ Display

- ◆ Video Games highly depends on what user see or view. So always ensuring the user does not see anything he is not meant to see and remains engaged. The only display device is a mobile screen.

➤ Sound

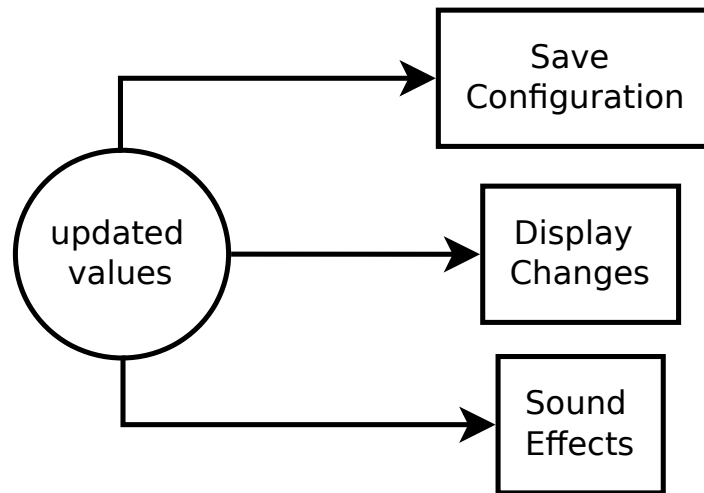
- ◆ Sound is an important aspect of video games. Hearing immediately improves the user experience. Some of the sounds used are listed as:

- button click
- engine
- destroy
- fire
- music

➤ Configuration file

- ◆ Fresh installs do not have configuration files. So creating a file if not already exist and updating that on new values. For example new high score.

➤ Below Image explains output design



4.3 Code Design

The coding design approach is Object-Oriented. For this purpose we used Gd-script main considerations are:

- Object-Oriented Approach
- Interaction of components
- Design patterns
- Algorithms

4.3.1 Object-Oriented Approach

Since I am using a game engine and Gd-scrip is embedded into the engine so I didn't have to use classes on my own I just created separate scenes which act as separate class and then call them where ever I need.

4.3.2 Interaction of components

All the components are highly dependent on each other. So their interaction must be ensured, and to achieve this Global.gd script is created which is accessible by all script files. Interact-able changes are written in Global.gd and other files automatically updates with the updating values in Global.gd because they directly use those variables.

4.3.3 Design Pattern

All the object creation technique in the game almost follows factory method. Not exactly factory method because Some changes were required to meet the requirements and due to limitations of Gd-script.

4.3.4 Algorithms

Most work is done procedurally which required good algorithms for example fire spreading is procedural you can change a variable value to get 100 fires or 3000 fires and they all will spread accordingly. Besides a big part of code is being reused so again using algorithms reduced total number of lines and made maintenance easier.

4.4 User Interface Design

A video game highly depends on user interface. We have tried to achieve the following golden rules while designing the interface of our project.

- Place the user in control
- Reduce User memory load
- Make the interface consistence
- Keep the resolution optimized for each objects

Interface consists of following components and they are in no particular order:

- Splash screen
- Main Menu UI
- Main Scene

! “[Please refer to [user guide](#) for images]”

➤ **4.4.1 *Splash screen***

- Very first thing user see at start up is this splash screen.

➤ **4.4.2 *Main Menu UI***

- The main menu screen where user can choose to perform various actions and configure game options, like difficulty, ship, mode, etc.

➤ **4.4.3 *Main Scene***

- Once user hit play button this scene is loaded into memory. It includes.
 - background
 - Health bar

- life indicators
- pause button
- return button
- score
- player
 - engine fire
- player fire
- levels main
 - enemies
 - virus
 - todda
 - engine fire
 - fighter
 - engine fire
 - enemy fire
 - boss
 - boss fire

5-Game Development

Outline:

1 Customer Requirements

2 Development method

2.1 Choosing game engine

3 Planning the game

3.1 Game Art

3.2 Programming the game

3.3 key features

4 Publishing the game

The following chapter will discuss the development of the MilkyWay the mobile game. I will talk about the development phase by phase and also discuss about game arts, programming and publishing the game after development phase complete.

Development of the game started in February 2021.

5.1 Customer requirements

In February 2021 the concept stage had started with gathering the requirements from the customer. This was done by face-to-face meetings, telephone conversation etc.

After the requirements were gathered, we mapped the target audience and the game requirements.

“Kids always loves what is made for kids older than their own age group. For example, an 8 - year- old kid wants to play a game that is made for a 10 - year- old kid. And a 10 -year- old kid wants to play a game that is made for a 13 - year- old kid. A 13 - year- old kid wants to play a game that is made for an 18-year-old. Most of the kids aren’t interested in playing games which are directly targeting their age range.” (Rogers, S. 2015)

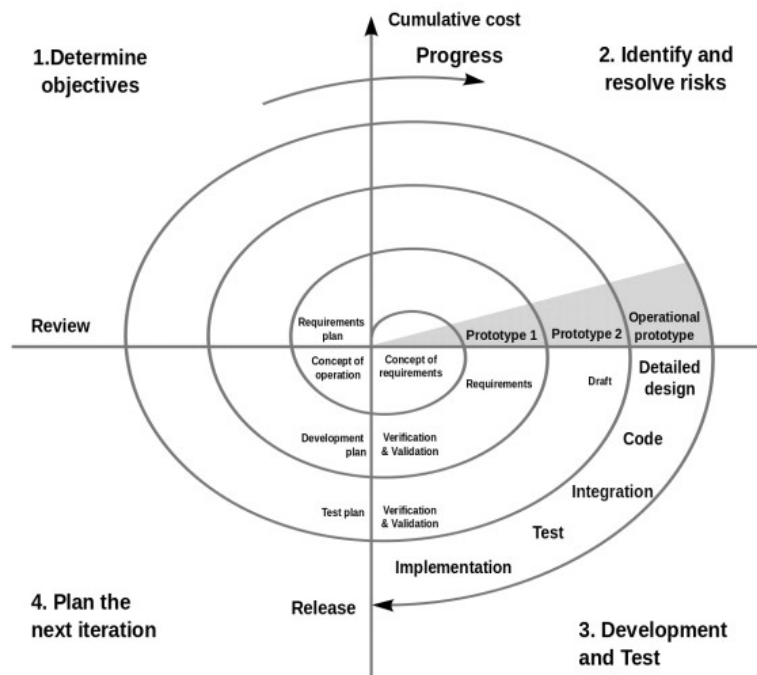
the MilkyWay mobile game is for children age 5 and up the most important thing for the game was that it had to be fun. This is done by creating colorful and creative game arts, easy controls and happy music.

The main feature of the mobile game is it is a completely offline mobile game. So, it can be used without internet and also it is design for less specification mobile phone like (1GB or 2GB RAM mobile within older Andriod Version like KitKat).

5.2 Development method

We propose build and fix model for prototyping very first linear build of the game and then rewriting the code using Object-Oriented Programming and Design (OOP&OOD)

SPIRAL software process model of implementation of MILKYWAY offline game for children's entertainment.



The Spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The Spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.

A software product repeatedly passes through these phases in iterations. The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral.

5.2.1 Choosing a game engine

We had to choose what game engine we would use for the development of MilkyWay mobile game. The game engine had to support all platforms and be flexible to allow future expansion on the mobile game. After a comparison between game engines, we had decided to use Godot for its cross-platform support and It is an open source as well. Godot has many benefits and features that would support us in the development of the mobile game. The main benefit of using multiple platform support, was that play testing and showing game demos become much easier.

5.3 Planning the game

Now we discuss the planning of the development project. The design phase included benchmarking MilkyWay game for mobile devices. This is to view the games from a player perspective of the feel of the gameplay and what features other team members had added to their game to make it more enjoyable. When the planning stage had been completed the workload had to be equally divided into categories and completed one at a time.

The development was split into Art which contained all graphical work, the menus, the cinematics, and animation work for the game. The programming of the game included player behavior, objectives, scoring, spawning and deleting of game objects, and creating the entire work of the game.

5.3.1 Game Art

Game art is just as important as its other features.

The process to create different arts for the game was long and hard.

We had to work hard for this and after a lot of hard work we made such arts that would make the children happy and increase their interest in the game. Here are the list of arts

Player's ships are:



s d

UI art



Enemies at different levels



5.3.2 Programming the game

The programming tasks were done by the lead programmer of the project. During the design phase the team decided to create the game based on an existing template to speed up the process. This decision saved time on programming of the game, however the risk in this was that errors and bugs had to be investigated and would take a lot of time to fix.

The programming of MilkyWay mobile game was split into two categories the main game logic and the boot and menu of the game. The main logic of the game consisted of the player movement, player controls and the drawing of game objects.

- First, we define some global variables
 - ◆ player max health
 - ◆ player index
 - ◆ player damage
 - ◆ enemy damage
 - ◆ enemy max health
 - ◆ high score
 - ◆ screen resolution
 - ◆ game mode
 - ◆ game speed
 - ◆ game speed factor, etc.

- Second, we setup the main menu screen
 - ◆ Selector for game mode
 - ◆ Selector for player's ship
 - ◆ Selector for difficulty
 - ◆ button for volume
 - ◆ label for high score
 - ◆ Set the values and code their behavior

- Third, we code Players
 - ◆ Area2D type
 - ◆ add collision
 - ◆ sprite animation of destroyed
 - ◆ engine sound
 - ◆ add shaders
 - ◆ Code movement

- create a players_main scene
 - ◆ include all the players in that scene
 - ◆ code to return requested player

- Create a player fire scene
 - ◆ Area2D type
 - ◆ add collision
 - ◆ add visibility notifier
 - delete when not visible to screen for memory management.
 - ◆ Code behavior

- Create enemy fire scene
 - ◆ Area2D type
 - ◆ add collision
 - ◆ add visibility notifier
 - delete when not visibility to screen for memory management
 - ◆ Code behavior

- Code some enemies
 - ◆ Area2D type
 - ◆ add collision
 - ◆ sprite animation of destroyed
 - ◆ add sound effects
 - ◆ add shaders
 - ◆ code their AI behavior

- create a levels scene
 - ◆ include all enemies in the scene
 - ◆ add their spawn timers

 - ◆ code their spawning behavior.
 - In story mode it should be as
 - level 1 (virus only)
 - level 2 (virus and todда)
 - level 3 (virus, todда and fighter)
 - level 4 (virus, todда and fighter with increased speed)
 - level 5 (final boss)
 - in endless mode it should be as
 - enemies (virus, todда and fighter)
 - increase speed after every 20 increment in current score.

- Create a main scene
 - ◆ include levels scene
 - get game mode from global script and behave accordingly.
 - ◆ include players main scene
 - get player index from global scrip and return the correct player
 - ◆ include player_fire

- on touch open fire
 - check for any power ups
 - spread fire accordingly

5.3.3 Key features

A good game requires some good features to make it interesting. We focus on three different categories in making a more enjoyable game for kids. The categories are challenge, sensation, fantasy. Each category is equally important in making a great game.

The key features of MilkyWay game is a flawless speed, especially designed low-quality mobiles, an offline game so, there is no advertisement no use of internet, consume very little storage, colorful arts, addictive music and sounds (sound form: kenny.nl).

5.4 Publishing the game

When the development of MilkyWay Mobile Game was complete the release of the game started. For this Purpose first of all we release game on Google Play Store and then advertise it on different platforms.

We all so invite different friends to download the game and play this game with there kids and sends us reviews.

6-System Testing

Outline:

- 1 Introduction**
- 2 Code Testing**
 - 2.1 Unit Testing**
 - 2.2 Integration Testing**
 - 2.3 Performance Testing**
- 3 Project Testing**
- 4 Project Reviews**

6.1 Introduction

The most important part of any game development process is the testing phase. Game testing plays a huge role in the development as it allows designers to test their work to test flaws and accuracy their own work.

“As a game designer, the first and last play tester of the games you design will most likely be

you. You will be the first person to try out each of your ideas, and you’ll be the first person

to decide whether the game mechanics and interface feel right.” (Reference: Gibson, J. 151)

To test our mobile game, we divided the testing process into three stages. There are,

- Code Testing
- Project Testing
- Project Reviews

6.2 Code Testing

At this stage, we test our code to see if it works. For this purpose, we have divided the code into different test cases which help us to test the code specific to a particular function.

6.2.1 Unit Testing

Since a video games have many modules and we use spiral win win so we tested modules separately some of tested modules are:

- Player0
- Player1
- player_fire
- final_boss
- virus
- enemy fire
- h_fire_power_up
- v_fire_power_up
- background_env

6.2.2 Integration Testing

Once all the units are tested they all are actually participating in a single module called Main_Scene so they must be tested for integration. For Example:

Enemy ships that fire must fire from the gun and once fire is spawned it should be independent of the ship and move straight as it should.

Power ups like h_fire_power_up and v_fire_power_up are just two area nodes emitting signal on collision with player and

procedurally adds more fires to the player's ship.

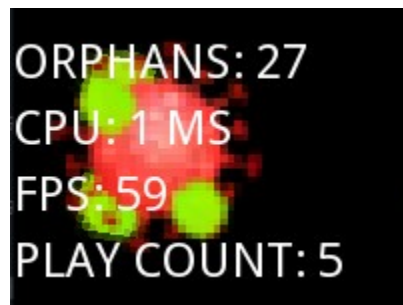
Ships comes with different shapes so h_fire_power_up must consider the shape while spawning fire.

Below table shows integration testing results;

Module	Module	Pass	Fail
Player	Fire	✓	
Fire	H fire power up	✓	
Fire	V fire power up	✓	
Fighter	Enemy fire	✓	
Final boss	Boss fire	✓	
Player	Life power up	✓	
Background environment	UI screen	✓	
Background environment	Main screen	✓	
Levels main	Main screen	✓	
UI	Global	✓	
Main scene	Global	✓	

6.2.3 Performance Testing

Since the video game claims to be performant on low-end Android mobile devices so it had to be tested for performance. Godot var type variables are 20 bytes and casting is also not good in Gd so I used PoolByteArray() for unsigned numbers less than 255 and same for integers and Real numbers which drastically improved performance of the game. Here are some in game statistics overlay screenshot. On emulation but m sure fps are pretty much same on actual devices as well. And no device outputted lower than 30fps.



[Note: this is picture of in engine and is not visible to user by default so you may need to make “visible=true” for some stuff and uncomment some lines to view these statistics of the game.]

6.3 Project Testing

At this stage we test our project (MilkyWay Mobile Game). For this purpose, we installed it on different devices and analyze the game installation process, game performance, game smoothness, game flow, and game behavior on different devices.

We did this at different stages of the development of the MilkyWay mobile game. And even after the development is completed. We repeated this process several times on different devices. After the test is complete, we compile a result that is below in the table.

Device	Test Name	Pass	Fail
Q-Mobile Noir S5 Android OS, v5.0 (Lollipop) 8GB built-in 1GB RAM	Installation Test	✓	
	Touch Screen Test	✓	
	Mute / Unmute Test	✓	
	FPS Test	✓	
	Game Flow Test	✓	
Samsung Galaxy Grand Prime Android OS, v5.1 (Lollipop) 8GB build-in 1GB RAM	Installation Test	✓	
	Touch Screen Test	✓	
	Mute / Unmute Test	✓	
	FPS Test	✓	
	Game Flow Test	✓	
Realme 3 Pro Android OS, v9.0 (Pie) 64GB built-in 4GB RAM	Installation Test	✓	
	Touch Screen Test	✓	
	Mute / Unmute Test	✓	
	FPS Test	✓	
	Game Flow Test	✓	

Sony Expria P (LT22i) Android OS, v4.0.4 (ice cream sandwich) 4GB build-in 1GB RAM	Installation Test	✗	
	Touch Screen Test	!	
	Mute / Unmute Test	!	
	FPS Test	!	
	Game Flow Test	!	

6.4 Project Reviews

Since this is a university project, we cannot publish it on the Play Store or other platforms. Therefore, for this purpose, we personally send this game to our friends, colleagues, and other professionals and request them to install this game on mobile phone play with their children. And send us feedback on game performance, game artwork, game features, and more. Also, tell us about the children's interest in the game.

Since then, we have received a lot of messages from different people.

7-Future Extension

Now we discuss the future extensions of MilkyWay mobile game. In the future, we may add several updates.

Some of them are following

- we can add various levels in story mode.
 - As there are only five levels in the story mode in this edition. In the future, we may add more levels to the game. There are only two players in this edition. In the future, we can add more levels to it.
- we can add more player
 - There are only two players in this edition. In the future, we can add more levels to it. It means there are more types of planes with different features are available to play.
- we can add more shaders and effects
 - In the future we can add different effects and shaders in the games. Which make the main outlook more attractive.
- we can add different Eventual modes
 - In the future, we may add various eventual modes to the game. For example, on February 27, 2019, the Pakistan Air Force destroyed an Indian Air Force jet and arrested a pilot named Abhi-Nandan. We can add different events like this. This not only increases the

knowledge of the children but also their interest in game.

➤ we can add various enemies

- In the future, we may add different types of enemies. In this edition, we have included an enemy in the form of corona virus due to this epidemic. In the future, we may add enemies with different shapes and characteristics. These enemies have different functions and different types of damaging tools.

➤ we can add art works

- Because this game is specially designed for children. Therefore, in the future, we can add different artworks and graphics to make the game interface more attractive. This increases the interest of children.

➤ we can add various fire type, power ups etc.

- We can also add different types of fires and power ups in the future. We now add three to four types of fire. In the future, we can add more.

8-User Guide

Outline:

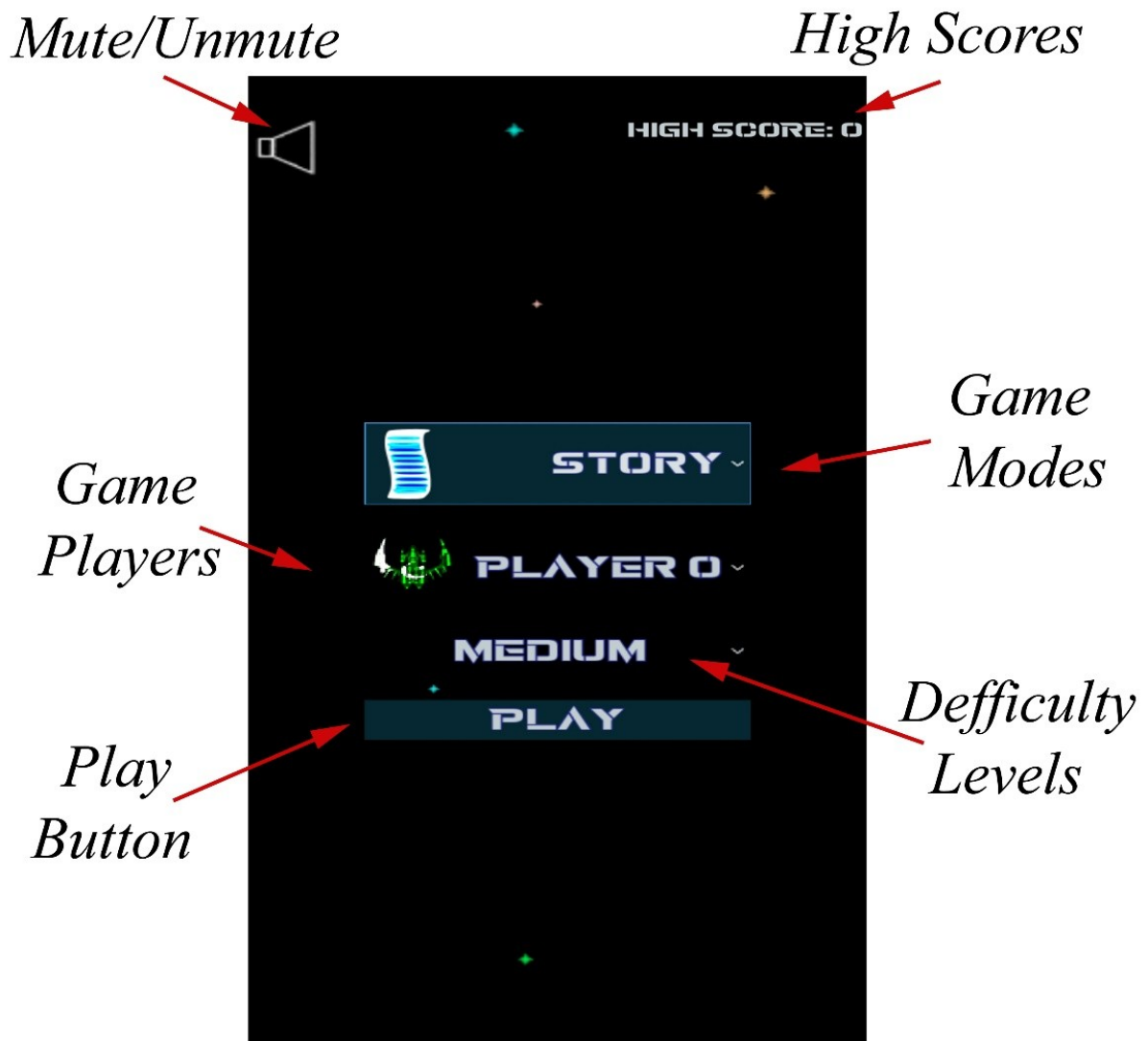
- 1 User Interface**
- 2 Game Modes**
- 3 Game Players**
- 4 Game Difficulty Levels**
- 5 In Game Interface**

This chapter is about user guidance. In this chapter we guide the user with the help of screen shots.

8.1 User Interface

This screenshot shows the user interface of MilkyWay Mobile Game. It shows the different buttons and modules.

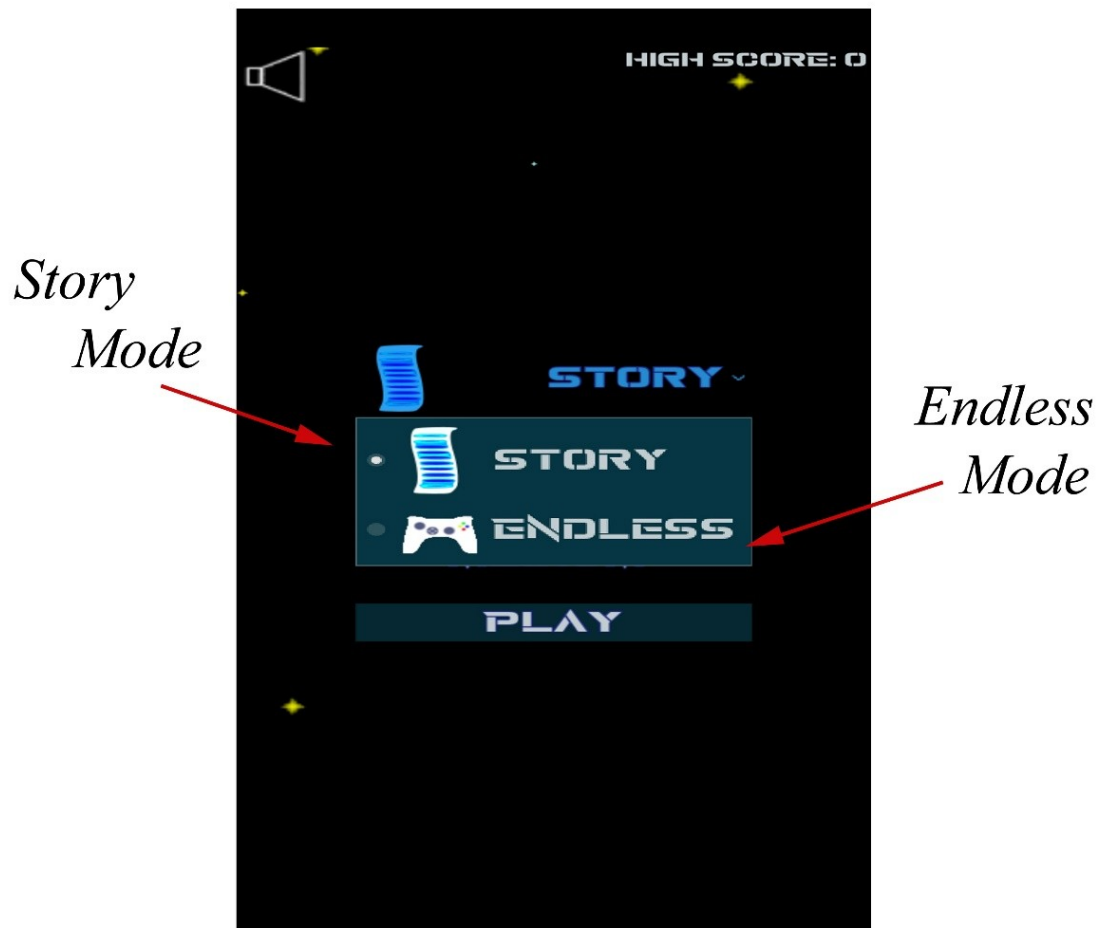
- In the upper right corner, there is a High Score indicator. It shows the highest score that a player gets in a game.
- In the upper left corner is a button to mute and unmute game sounds or music.
- In the middle, the first button is to select game mode. We've added two modes to this game. The first Endless mode and the second Story mode.
- In the middle, the second button is to select game player. We've also added two players to this game. The first Player0 and the second is Player1.
- In the middle, the second button is to select game difficulty level. Which are Easy, Medium and Hard.
- In the last, there is a play button to start the game.



8.2 Game Modes

The game has two modes, one is endless and the other is story mode. Endless mode is an unlimited game with no levels. In this mode, the difficulty increases with your score.

On the other hand, there are five different levels of story mode. The difficulty in the game increases, as you pass the level. There are enemies at every level. In the end there is a big enemy which we call boss enemy. You have to kill the boss to win the game.



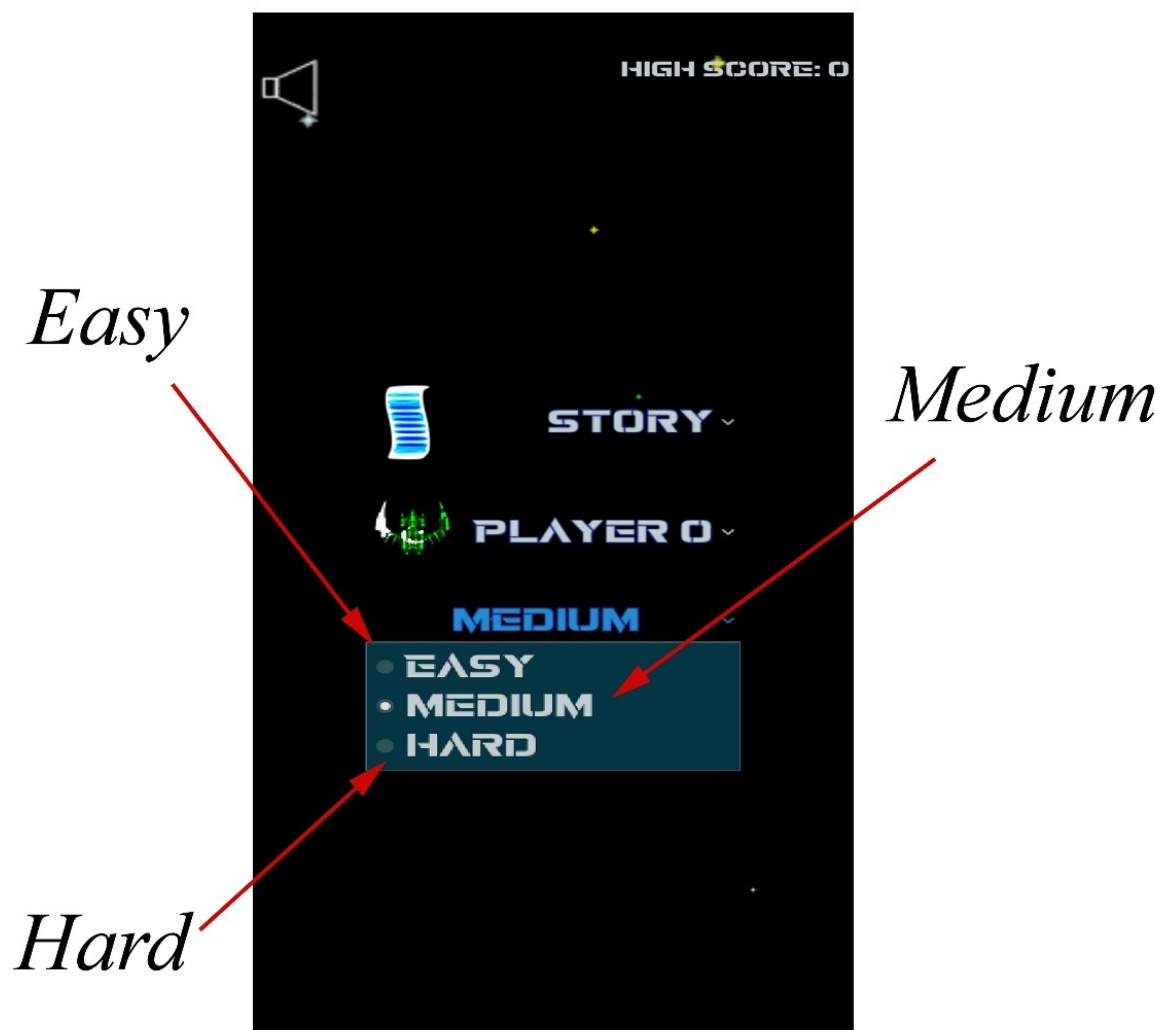
8.3 Game Players

There are two players in the game. These are player 0 and player 1. In fact, the players here are different designs of planes. As you can see in screenshot.



8.4 Game Difficulty Levels

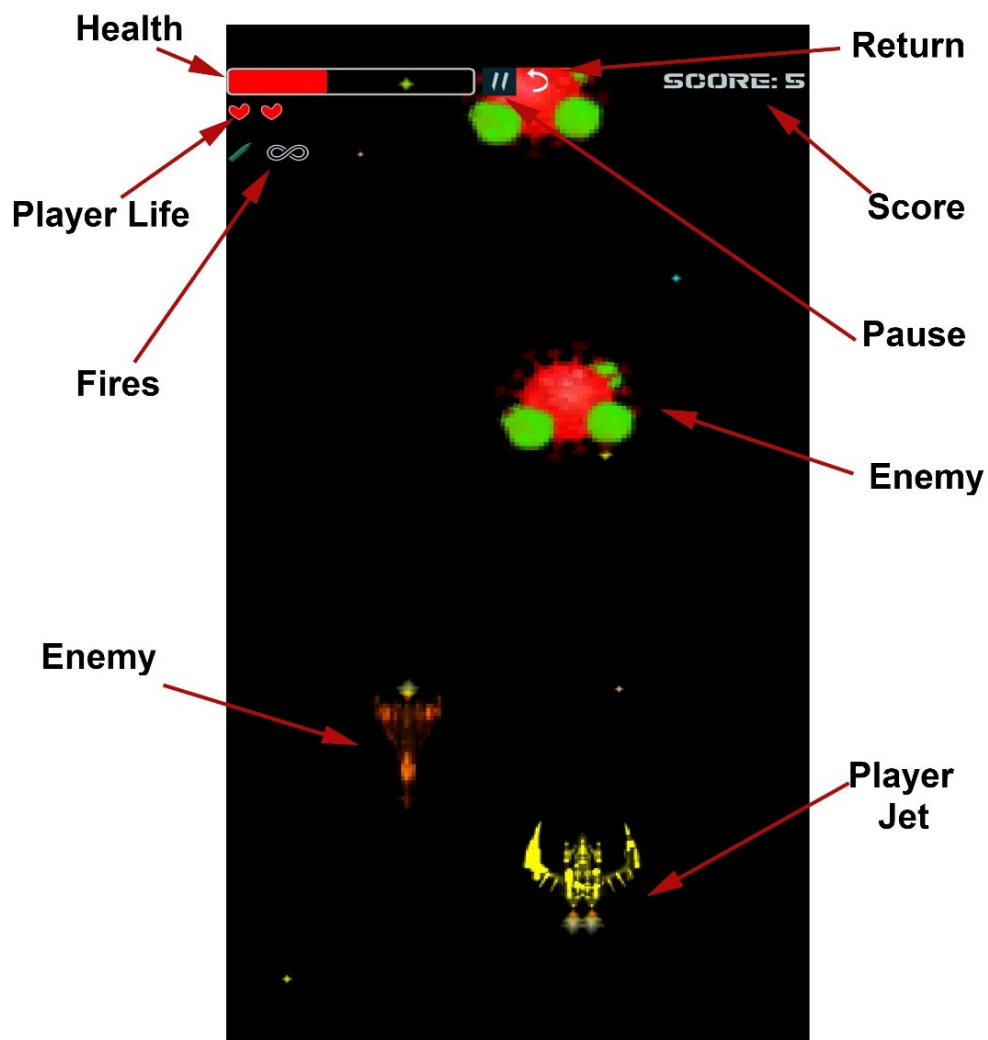
The difficulty level of the game is to choose the difficulty of the game. There are three levels of difficulty in the game. They are easy, medium and hard. You can choose the difficulty level at which you want to play.



8.5 In Game Interface

As you can see in the screenshot in the left-hand corner there is a health line, player's life and fire signs. On the next health line, there is a pause and a return button. And on the right-hand corner, there are scores.

In between, there is an enemy jet and a virus enemy. And below is your ship.



9-References

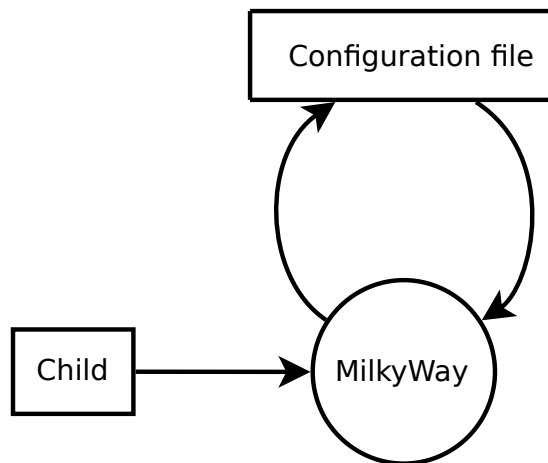
There are not much of references because No libraries were used everything is done with these core tools but they are listed as:

- [Godot](#) Game engine used for development purposes.
- [Blender](#) DDC for pictures and animations.
- [Krita](#) Painting program for artwork.
- [Google](#) search engine for references.

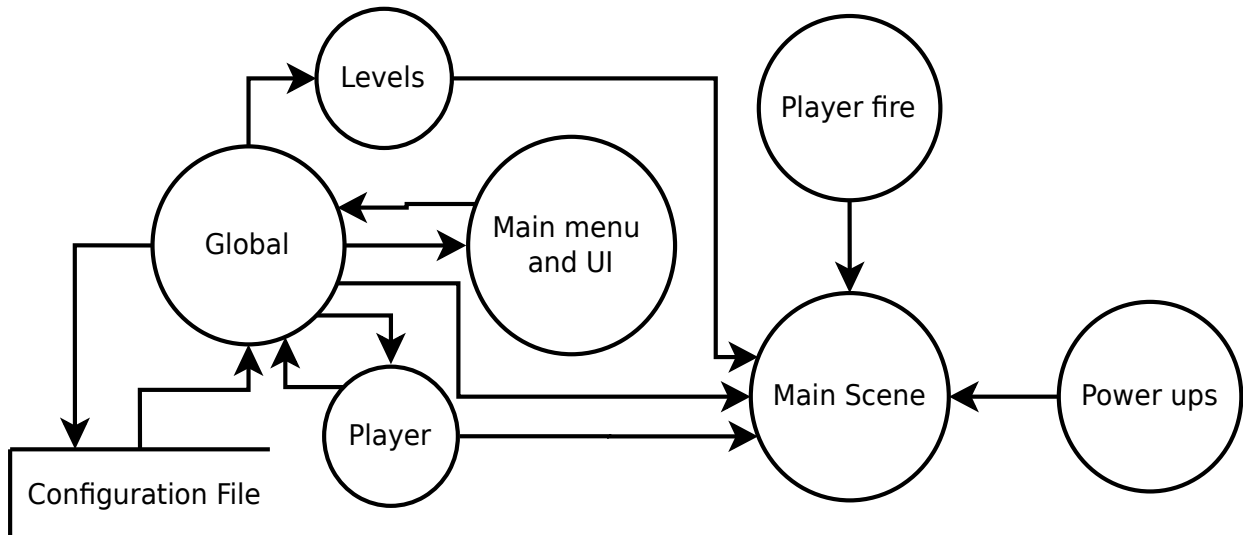
10-Appendices

No Database was used because only one file is being read or written. So no ERD or such things.

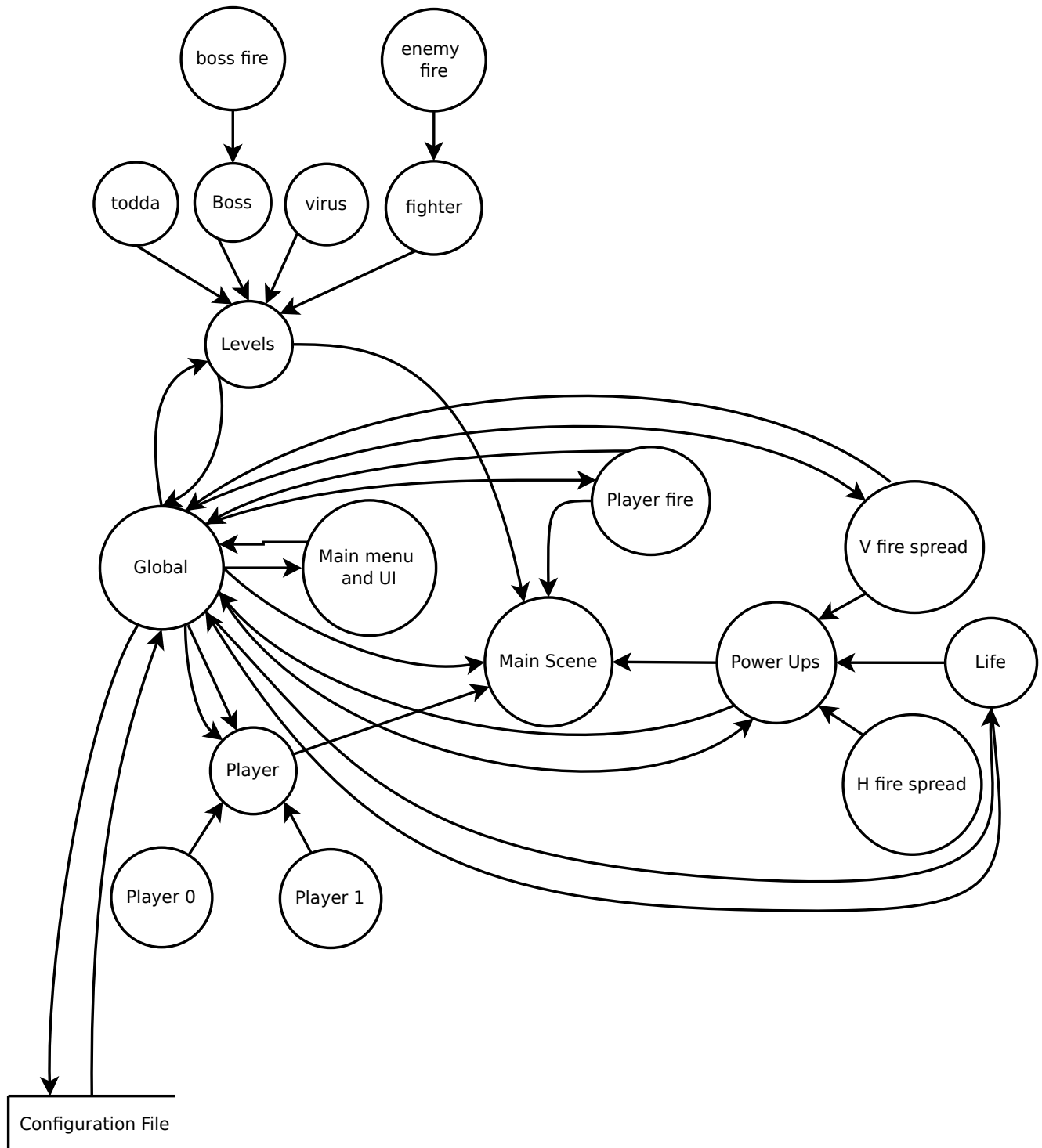
Level 0 DFD:



Level 1 DFD



Level 2 DFD



Flow Chart

