CS-319 Term Project

Defender

Analysis Report

Team:

- Hanzallah Azim Burney
- Abdul Hamid Daboussi
- Perman Atayev
- Gledis Zeneli
- Endri Suknaj

Instructor: Ugur Dogrusuz
Teaching Assistant(s): Hasan Balci

Contents

Introduction 3

Proposed System 3

Overview 3

Functional Requirements 3

1.2.1 Mothership 3

1.2.2 Landers 4

1.2.3 Baiters 4

1.2.3 Bombers 4

1.2.4 Humans and Mutants 4

1.2.5 The Landscape 4

1.2.6 Laser Weapon 5

1.2.7 Health 5

1.2.8 Waves 5

1.2.9 Map 5

1.2.10 Score 5

Non-functional Requirements 5

1.3.1 Scalability 5

1.3.2 Maintainability 5

1.3.3 Usability 6

1.3.4 Reliability 6

System Models 6

1.4.1 Scenarios 6

1.4.2 Use-Case Model 6

1.4.3 Object and Class Model 7

1.4.4 Dynamic Models 7

1.4.5 User Interface 11

References 11

Analysis Report

Defender

Introduction

Defender is based on the classic 1981 arcade game of the same name developed by Williams Electronics. It was the company's most popular game selling over 55,000 units and praised for its gameplay and audio-visuals at the time [1]. This desktop based version tasks a single player to control a spacecraft and prevent aliens from turning humans living on a planet into mutants. The player must try to save as many souls as possible from each wave of aliens.

Defender was chosen as the project since it provides a unique opportunity to delve into OOP principles and come up with a proper software engineering solution due to the various objects, interactions and hierarchies it contains while providing a simple enough game design to implement and use.

1 Proposed System

1.1 Overview

The first iteration of Defender is tightly coupled with the original version except for minor changes such as points assigned for various actions, graphics, types of aliens and the potential restriction of the number of levels to be played by the user. The game consists of the user opening the GUI and entering the main menu where he will have the option to start a new game, load a saved game (if any), change settings, check the high scores of the game, get help which would redirect to the online wikipedia page for the game and credits. When the user starts a new/saved game he will enter the game where he can keep on playing until he dies or runs out of the time allotted for the specific level. The gameplay revolves around the user controlling his "mothership" which he can move up, down, left and right to kill aliens that are trying to pick up humans from the ground and turn them into mutants with a laser weapon. The user earns points specified in Section 1.2 for every hit and alternatively loses his life if hit by an alien or dies upon direct contact with an alien.

1.2 Functional Requirements

This section describes the attributes of the game components.

1.2.1 Mothership

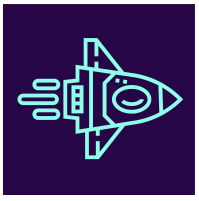


Figure 1.[2]

The mothership is the user controlled laser-equipped spacecraft that is controlled by the keyboard arrow keys to move in all four directions. It has a health level that decreases when fired upon by the aliens or when it hits a mine or is completely destroyed upon collision with the aliens. The use of keyboard keys rather than the mouse enables much more control over the spacecraft movement and allows the user to quickly adapt to the game.

1.2.2 Landers

Figure 2.[2]

This alien is the primary protagonist. It aims to infect humans on the ground and convert them into mutants by fusing with the humans. It has a laser weapon that shoots at the mothership.

1.2.3 Baiters

Figure 3.[2]

These are flat-shaped aliens that attempt to catch the mothership by matching their speed if they take too long to complete a level and fires accurate laser projectiles at the mothership.

1.2.3 Bombers

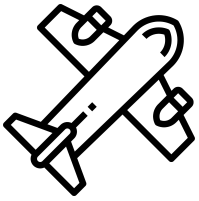


Figure 4.[2]

It is an alien that attempts to lay mines while being stationary. The mines reduce the mothership's life as detailed in Section 1.2.7.

1.2.4 Humans and Mutants

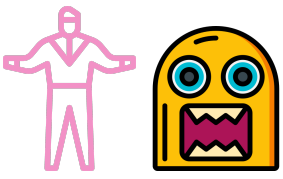


Figure 5 and 6.[2]

Humans are serve as the motivation for the game since, the user has to prevent them from being infected by the Landers. If the user fails then the infected humans and the respective Lander form a mutant which moves erratically making it difficult to gun down and attempts to chase the mothership while firing laser projectiles.

1.2.5 The Landscape

The landscape consists of a simple terrain denoted by an erratic line where the humans are placed.

1.2.6 Laser Weapon

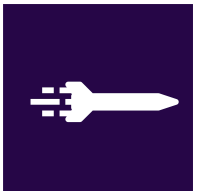


Figure 7.[2]

The laser weapon is used by both the mothership and the aliens, consisting in this iteration of infinite rounds.

1.2.7 Health

The mothership gets three lives, each equivalent to a 1000 points. A hit from a lander reduces this by 100, from baiters and bombers by 200 and from mutants by 300. Once the lives deplete, the mothership get obliterated. Direct contact with any alien being reduces one complete life of the mothership. On the other hand, the aliens are obliterated by a single hit from the mothership.

1.2.8 Waves

The alien waves signify a new level and are constrained by time. Initially the user gets 5 minutes to complete the level and the time reduces by 10 seconds with each succeeding level. The minimum time for the final level is 1 minute.

1.2.9 Map

The map at the top of the GUI shows a miniature version of the whole landscape indicating the positions of the mothership, aliens and humans. The map allows the player to better traverse the game and decide on which direction to go in.

1.2.10 Score

The user scores points for killing off aliens. For landers he gains 100 points, for bombers 200 points, for baiters 300 points and for mutants 400 points. If the score is in the highest top ten scores then it is saved to the highest scores list in the game which can be accessed by the main menu.

1.3 Non-functional Requirements

1.3.1 Scalability

The game will be constructed in accordance with OOP principles such as inheritance, polymorphism, abstraction and encapsulation to allow for the improvement, enhancement and extension of current features and integration of new features without changing much of the legacy code.

1.3.2 Maintainability

The game will be designed keeping in mind code efficiency and cleanliness to allow the current game implementers and other programmers to easily understand the code and use it as necessary.

1.3.3 Usability

The aim is to deliver an extremely user friendly product that the end user can enjoy playing for hours. Therefore, the goal is to keep the UI and controls simple following the normal conventions seen in similar implementations of the game, interactive and responsive to create a remarkable gaming experience.

1.3.4 Reliability

Game and scoring records will be kept within the local memory and the game will be an offline single player game to allow for reliable access in case of factors such as poor internet access. In terms of the code, it will be kept highly modularized and reusable to ensure the components work well.

1.4 System Models

1.4.1 Brief Scenarios

1.4.1.1 Play Scenario 1

The player starts a new game. The player controls the mothership by moving in all four directions to avoid collision with an alien or avoiding getting hit by a projectile. The player shoots and hits an alien earning points in the process depending on the kind of alien hit. If the player shoots a human no change occurs. However, the player fails and collides with an alien that kills the user instantly and ends the game.

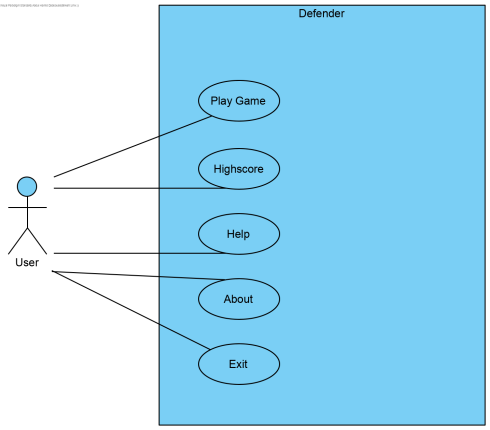
1.4.1.2 Play Scenario 2

The player starts the game and moves the mothership around avoiding aliens and hostile projectiles. However, the player fails and a projectile form an alien (except bomber) hits the mothership leading to a decrease in the motherships health.

1.4.1.3 Play Scenario 3

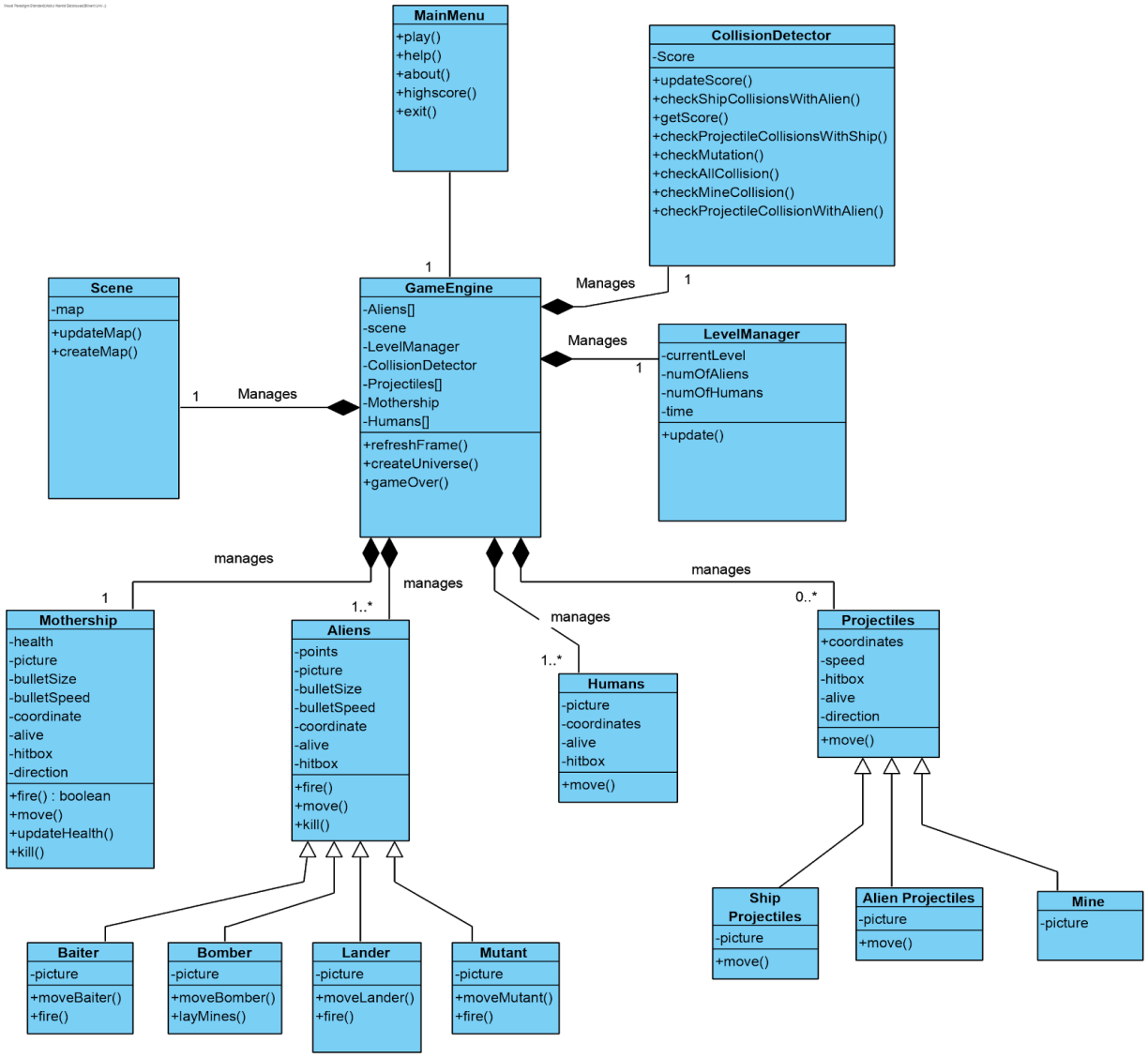
The player starts the game and moves the mothership around avoiding aliens and hostile projectiles. The player is successful in doing so and in the process the player shoots projectiles killing aliens, saving humans and earning points. When the time runs the score, health and the mothership are carried forward to the next level with reduced time. The game can also end in victory if the user simply returns to the main menu by selecting the exit option from the screen.

1.4.2 Use-Case Model



1.4.3 Object and Class Model

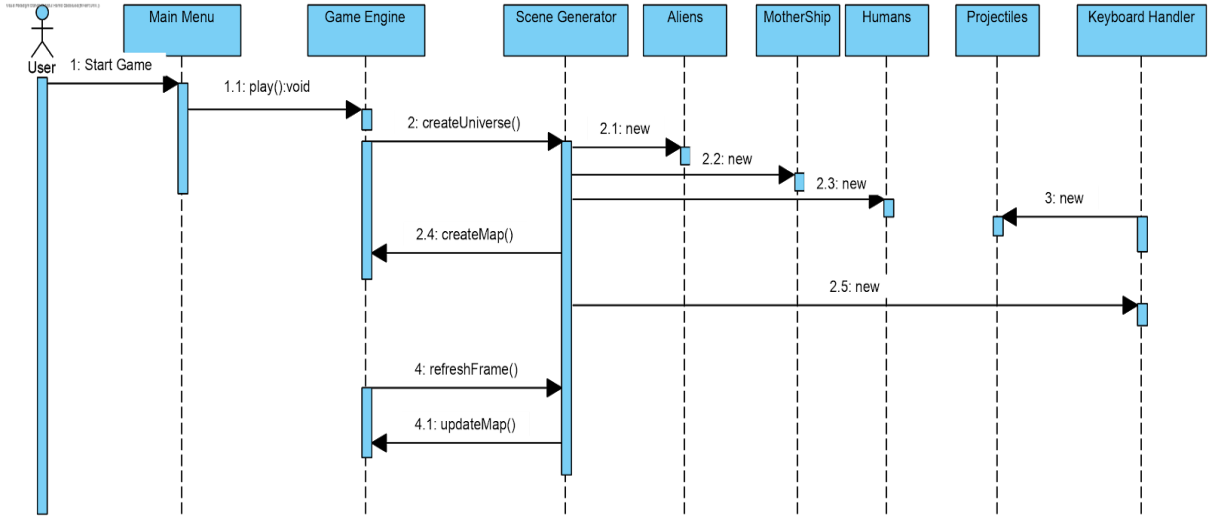
The class model for the game with the corresponding classes, methods and attributes is as follows,



1.4.4 Dynamic Models

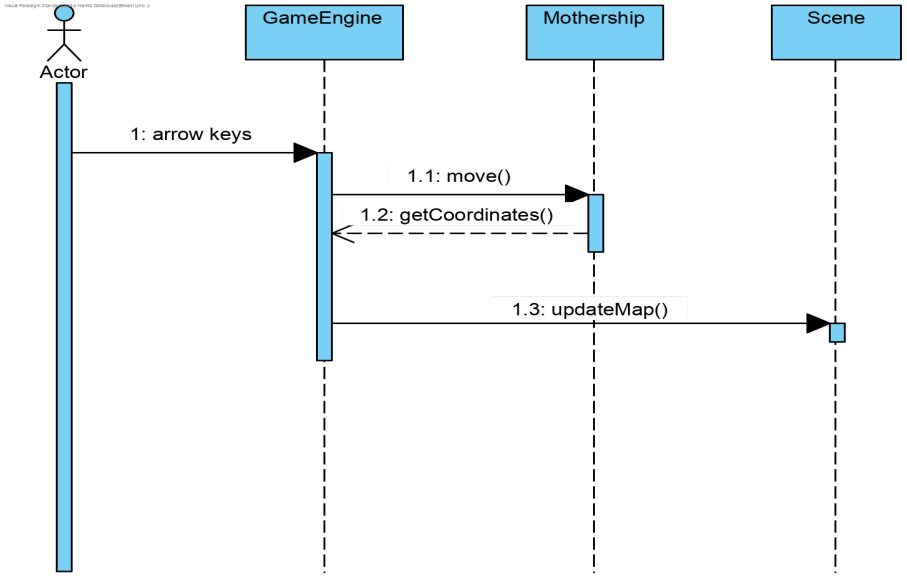
1.4.4.1 Start Game

The user clicks play game from the main menu and starts the game. This informs the game engine to create instantiate the scene generator and create the whole game infrastructure from the map to the score and health displays to the instances of the mothership, aliens and humans. The scene generator also connects to the keyboard handler detecting any change such as a projectile being shot and then creating instances of the projectile and then refreshing the whole game infrastructure.



1.4.4.2 Move Mothership

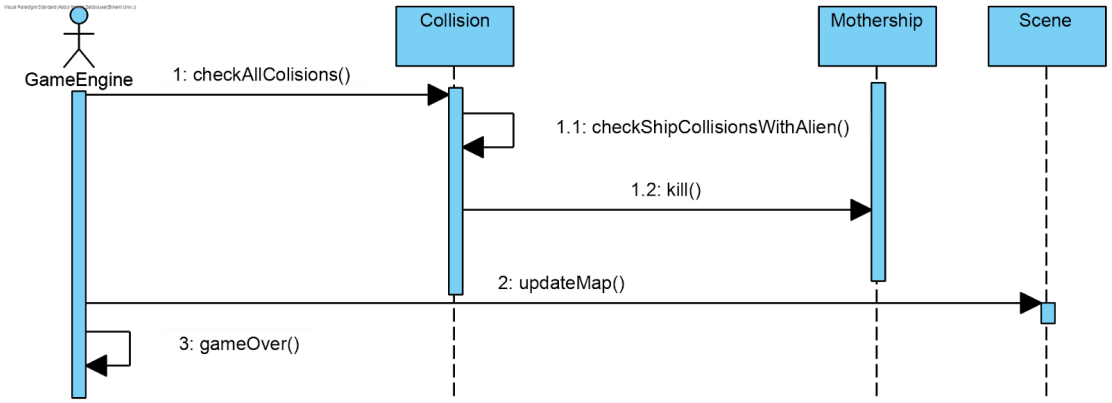
The mother moves along the (x,y) coordinates using the arrow keys and the updated coordinates when the arrow keys are pressed is sent to the scene generator which then refreshed the game display.



1.4.4.3 Physical Collision

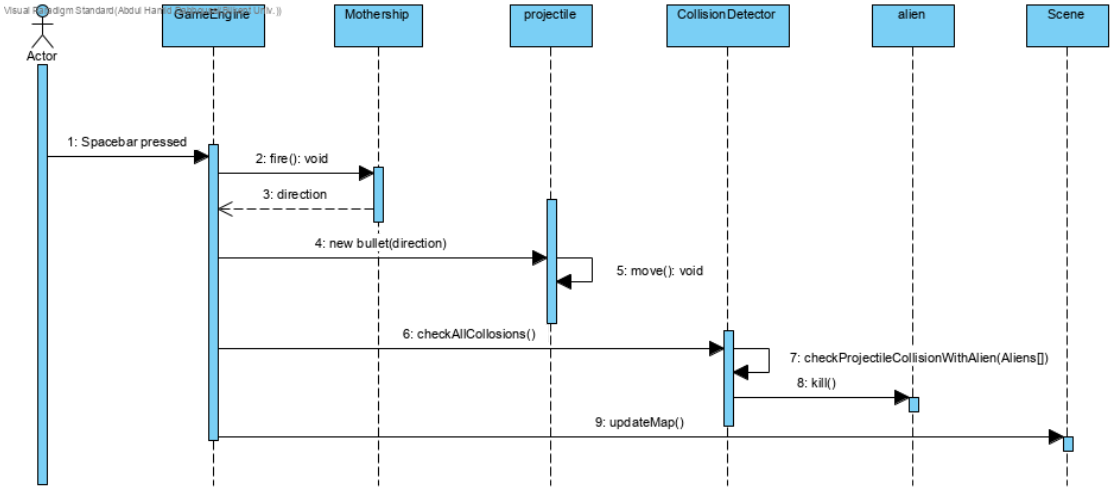
This sequence diagram is for checking collisions between the mothership and the aliens. The game engine constantly checks for all collisions form the collision detector class. If it detects a

collision of an alien with the mothership the game engine then terminates the mothership and updates the map. It finally calls the gameOver() method and ends the game.



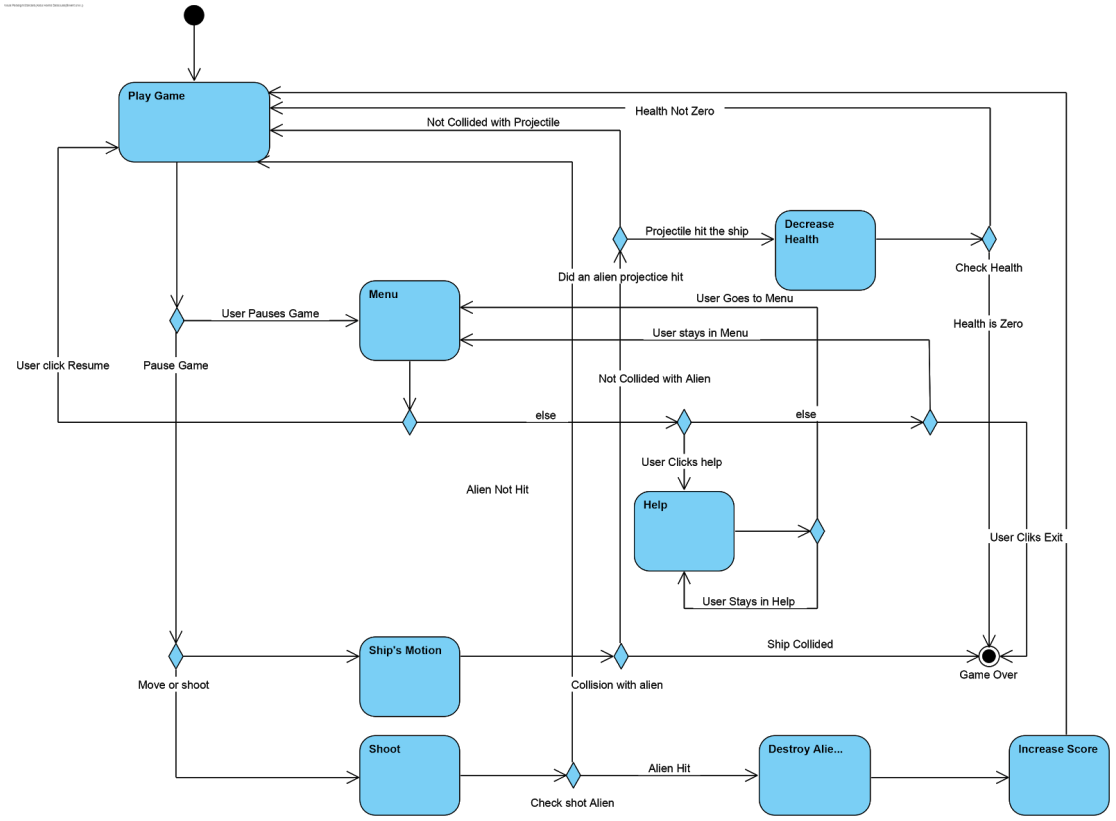
1.4.4.4 Shooting a Projectile

The main gameplay of Defender consists of the mothership's ability to shoot projectiles thereby killing aliens and by extension saving humans. A projectile is instantiated in the game when the spacebar is pressed. The game engine then moves the projectile and simultaneously initiates the fire method from the mothership to indicate that a projectile has been fired. The game engine then checks for the projectile colliding with any alien form and if there is a collision it then kills the alien in question. During all of this process the scene is being constantly updated to reflect the above changes.



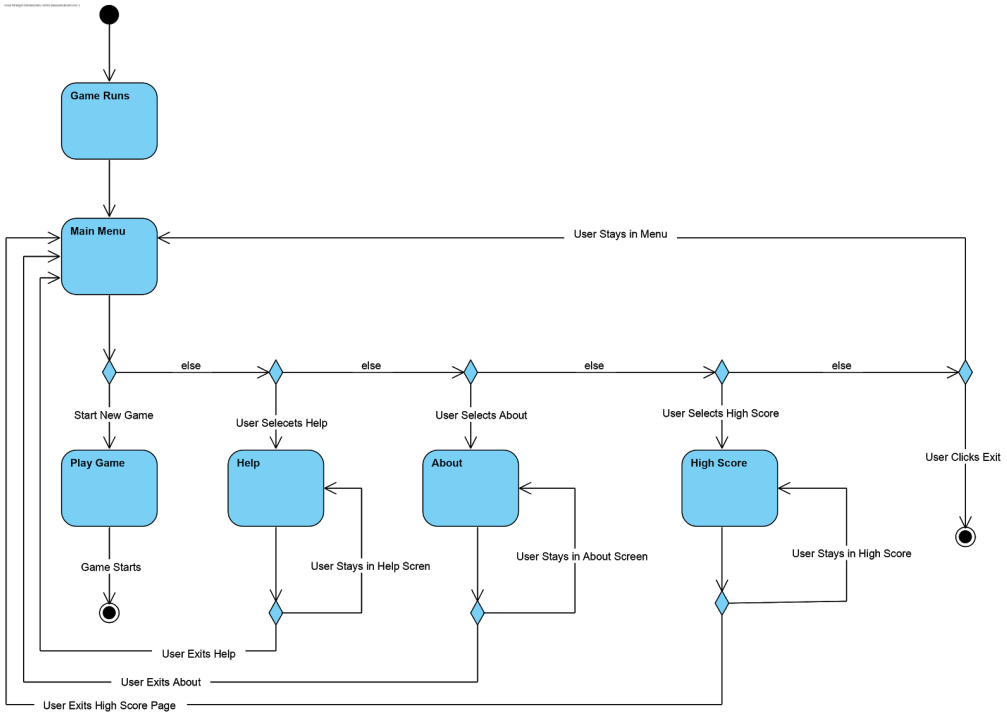
1.4.4.5 Play Game Activity Diagram

The following shows an activity diagram representing the game play. When the game starts, if the player does not pause, they have two options: moving the spaceship or shooting missiles. Pausing the game gives the user a menu that allows them to exit or access help. If the ship moves and hits an alien, the game ends and the player loses. If the missile shot hits an alien, the alien dies and the play gets points.



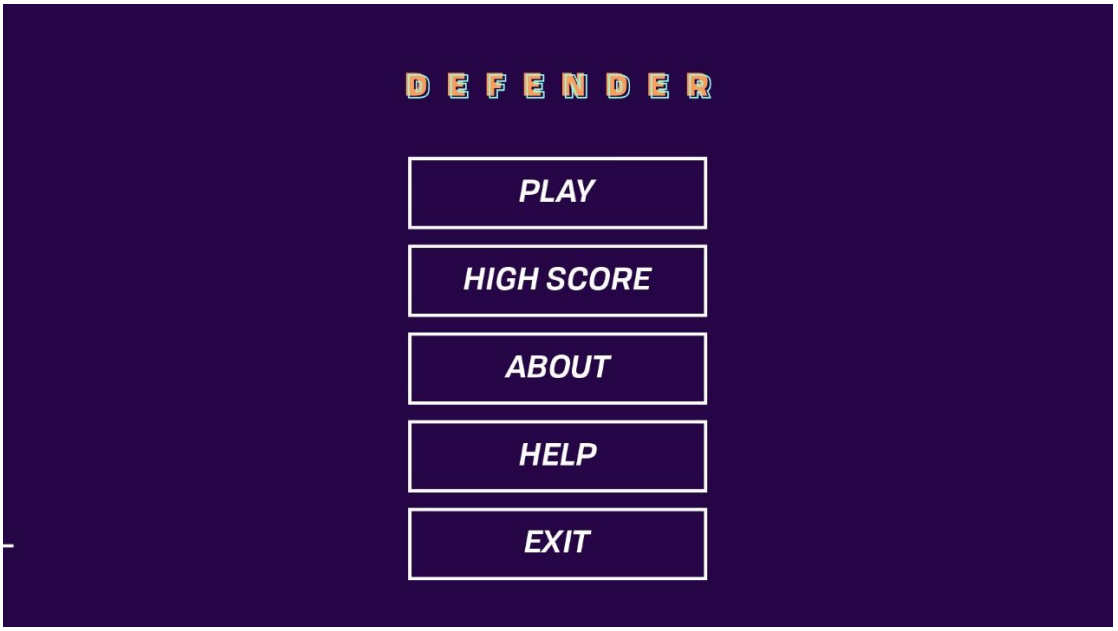
1.4.4.6 Main Menu Activity Diagram

The main menu is the page the user is greeted with then they first enter the game. They are provided with the option to play the game, access the help screen, about section, highscore list or exit the game.

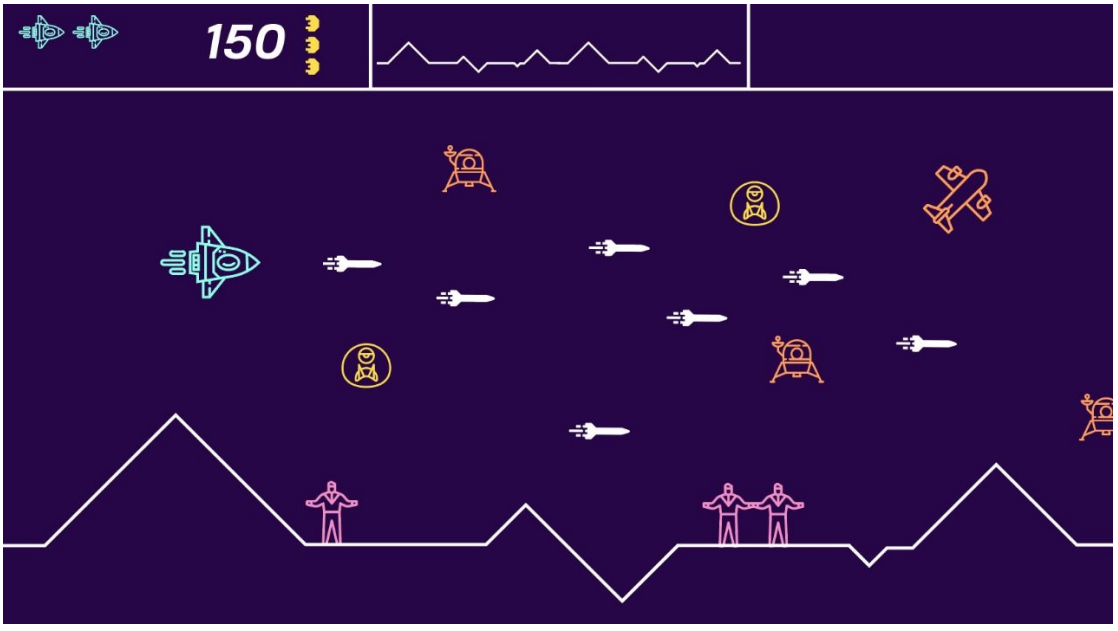


1.4.5 User Interface

The following diagram shows the main menu frame of the game. The UI is minimalistic and modern yet it draws upon the original arcade game's display for inspiration.



A mockup of the in-game UI is given below. Again, it is a minimalistic, flat and modern UI. Even though the design is modern, it is without a doubt based on the original 1980s game and takes design cues from the old arcade UI.



2 References

[1] "Defender (1981 video game)," *Wikipedia*, 18-Sep-2019. [Online]. Available: https://en.wikipedia.org/wiki/Defender_(1981_video_game). [Accessed: 10-Oct-2019].

[2] Flaticon. (2019). *Free vector icons designed by Freepik*. [online] Available at: https://www.flaticon.com/authors/freepik [Accessed 24 Oct. 2019].