BILKENT UNIVERSITY

CS 319 – Object-Oriented Software Engineering
Analysis Report
Intergalactica
Group 18


Ahmet Burak Şahin
21301755


Melis Kızıldemir
21401184


Selin Erdem
21401244


Ömer Sakarya
21301535

Table of Contents

1. Introduction

Intergalactica is a 2D arcade and adventure game that is similar to, yet more advanced than *Galaga* [1]. The main purpose in the game basically is to destroy all incoming alien invaders at each stage by using the gun that is attached in front of the user-guided spaceship.

INTERGALACTICA

Our logo for the game.

The primary criteria in our project in differentiating from other Galaga-based games is having a rich set of options and offerings. As will be described in more detail below in the overview section, *Intergalactica* enables user to practice an adventurous and diverse gaming experience compared to other versions in the field.

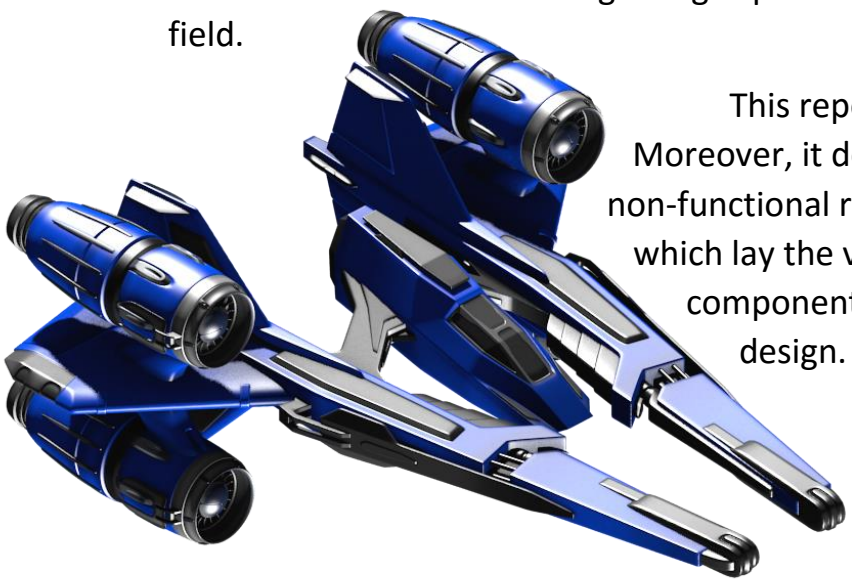


Image by MillionthVector [2].

This report covers an overview of the game. Moreover, it describes its functional requirements, non-functional requirements and use-case models which lay the very foundation of analysis components required to build a solid software design.

2. Overview

Opening Narrative: *Residents of Vitae are living in exile since the invaders from Galion 2, Xaves, Romia and Aduron took their home centuries ago. For a decisive victory, we need to contend all enemy presence throughout our path. The day of reckoning has come! For Alien Invaders!*

Briefly, as narrated above, the game consists of 5 levels/planets, each of which consists of 3 stages of alien attacks. A stage is over if an only if all the hostile alien presences are eliminated. The main objective in the game is to reach to the homeland (Vitae) without getting destroyed.

Level: 1
Planet: Galion 2

Level: 2
Planet: Xaves

Level: 3
Planet: Romia

Level: 4
Planet: Aduron

Level: 5
Planet: Vitae

Images by MillionthVector [2].

At each planet, clashes with aliens occur in the following stages: *Numbers vary with levels.

- ***First Stage:*** 20-16 *light* alien spaceships.*
- ***Second Stage:*** 20-16 *light* + 2-3 *medium* alien spaceships.*
- ***Third Stage:*** 20-16 *light* + 4-5 *medium* + 2-3 *heavy* alien spaceships.*

The spaceship which user controls can only move in either +x or –x direction. However, aliens can move in xy-plane unrestrictedly. In addition to movement capabilities of aliens, any alien ship can go out of sight from the sides. Meaning that if an alien disappears from the right-side of the screen, it is ensured that the alien will come from the left side of the screen, or vice versa.

The game ends either player makes it through the planet Vitae or gets destroyed.

Health points (HP) of player will be shown as a little bar beside the spaceship.

2.1 Alien Spaceships

Strength and agility of aliens depend on the level at which the user is.

Furthermore, the appearance of aliens also changes since each new planet corresponds to a new alien race. For example, Xavesian alien spaceships look like the following:

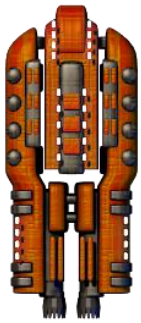
Class: Heavy

Class: Medium

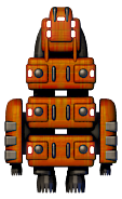
Class: Light

Romian ships look like:

Class: Heavy

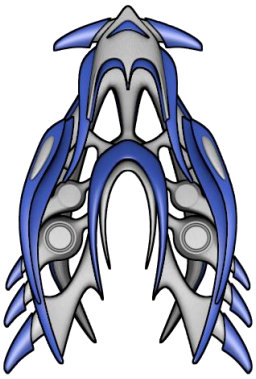


Class: Medium



Class: Light

Galion 2's are:



Class: Heavy

Class: Medium

Class: Light

And finally Aduron's are:

Class: Heavy

Class: Medium

Class: Light

All spaceship images by MillionthVector [2]

When an alien spaceship's ***health point (HP)*** reaches 0, it will get destroyed. HP will be shown as little bar beside alien.

Ships of Vitae, as being the toughest level, will be the mixture of the factions described above.

2.2 Item Drops & Power-Ups/Downs

When an alien spaceship gets destroyed, any of the following items can drop:

- **Power Ups:** Gives user an advantageous attribute
- **Power Downs:** Gives user a disadvantageous attribute
- **Money:** Ranging from 50-300
- **Gun:** There will be couple of gun types that can be either bought from the market or be dropped from an alien
- **Bomb:** Severely damages user's spaceship
- **Gem:** Gives points randomly in the range of [100, 1000000]

Power ups are as follows:

- **Shield:** Makes user's spaceship invincible for 20 seconds.
- **Extra HP:** Gives extra HP to the user
- **Extra Speed:** Increases the speed ratio.
- **Extra Fire:** Increases the rate of fire.
- **Score Multiplier x2-x5:** Multiplies points earned from aliens by either x2 or x5
- **Money Doubler:** Doubles the money

Power downs are as follows:

- **Gun Down:** Decreases gun to level 1
- **X-axis inverter:** Inverts the x axis, which makes controlling confusing.
- **Bomb:** Severely damages user's spaceship
- **Money Sucker:** Takes all of the money collected.

Power-ups or downs will be depicted as "?" sign, so user will have 50% chance of having either power-up or down.

2.3 Market Place

At the end of each level, the spaceship will visit the market place where user can buy:

- Power ups
- New Spaceships
- New Guns

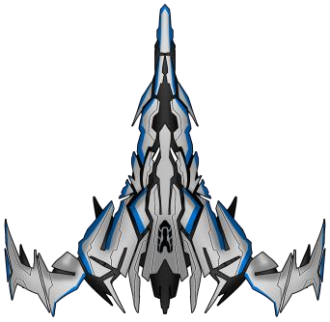
according to the money collected on previous levels.

2.4 Spaceships

At the beginning of the game, user will have this ship as default.

Image by MillionthVector [2].

As described above, if player collects enough money, he/she can purchase:



1st and 2nd Images by MillionthVector [2].

Image by Darke [3].

The difference between those ships are in their:

- Fire ratio
- Agility ratio
- Health Points
- Power ratio

Images above are shown in their order of strength from left to right.

2.5 Scoring System

During combat, when user manages to destroy an alien ship, user will gain varying amount of *points* according to which faction's ship was destroyed and what the class it was. For instance, a **large** class ship in *Level 1* (Galion 2 planet) will deal more points than the **light** or **medium** ones, ***or*** a **medium** class ship in *Level 3* (Romia planet) will deal more points than a **medium** ship in *Level 2* (Xaves planet), etc.

Aside from earning points by destroying alien ships, user can earn points via collecting *gems,* **if**, by a chance, drops when an alien ship gets destroyed. Moreover, if x2 or x5 score multiplier was active during the accumulation of points, the point earned will be multiplied accordingly.

When the game ends either by reaching to Vitae and winning all battles or getting destroyed by an alien, user will be prompted to enter his/her name in order to save the score made. In the main menu, there will be a *Highscores* option that will display all scores that have been entered so far, with respect to their magnitudes.

Every game session will have a top bar displaying the highest score achieved so far and the score that user currently has in the game.

2.6 Health Management

Every spaceship in the game has its own *Health Points (HP),* when a spaceship gets hit, the power rate of the attacker will be calculated and subtracted from the attackee.

2.7 Ship Properties

Every spaceship on the battlefield has:

- Agility Rate
- Fire Rate
- Power Rate
- Health Points

Agility Rate of a spaceship is the measure how quickly the ship can move in a given time. Agility rate can be increased by either purchasing or collecting the power-up: **Extra Speed.**

Fire Rate of a spaceship is the measure how frequent can shoot in a given time. Fire rate can be increased by either purchasing or collecting the power-up: **Extra Fire.**

Power Rate of a spaceship is the measure of how much a shot can deal damage at a time. It is calculated as

$$Power\ Rate = (Power\ Rate\ of\ Gun) * (Power\ Rate\ of\ Spaceship)$$

Power rate can be increased by either purchasing:

- A better gun
- A better spaceship

Health Points (HP) of a spaceship is the maximum amount of damage a spaceship can bear in a single hit. In order to recover HP, user can either purchase **Extra HP** from market, collect **Extra HP** power-up from a drop of an alien or wait as HP recovers itself at a rate of 2 HP/sec.

3. Functional Requirements

User will be able to:

- *Control the game via mouse and keyboard.*

During game session, user will be able to move the spaceship via left and right arrow keys and shooting will occur via CTRL key as default.

- *Enter highscore.*

When a game ends either by completing the level 5 or by getting destroyed, user will be prompted to enter a name which will eventually be recorded and associated with the score he/she made.

- *Change the settings of the game*
    - *User will be able to change key settings*
    - *User will be able to mute the game*
- *Access help menu.*

Help menu will provide basic tutorials regarding the in-game functionality.

- *See all highscores so far.*

Every time Intergalactica starts-up, it will fetch and restore locally stored highscores data which is a record of previous game scores.

- *Pause or Restart the game.*

Associated with a default key, in this case ESC, user can pause or restart the game whenever the button is pressed.

- *See credits.*

4. Non-Functional Requirements

Intergalactica will be:

- ***Accessible***

Our game will be accessible and free-to-use for anyone.

- ***Well-documented***

We will sustain the documentation quality as high and sufficiently detailed as possible.

- ***Efficient***

Overall efficiency of the game will be as high as it would not affect the gaming experience.

- ***Maintainable***

The game will be developed in a

- Readable
- Understandable
- Testable

fashion and thus will be highly maintainable.

- ***Portable***

Since it will be implemented in Java, any computer that has JVM will be able to run it.

- ***Reliable***

By taking into account every pitfalls and possible crashes, we will try to develop highly fault tolerable software.

5. System Models

5.1 Use-case Models

Use case 1:

Use case name: `PlayGame`

Participating actors: `Player`

Entry Condition: Options are being displayed to `Player`.

Exit Condition: The game has finished session.

Main Flow of Events:

1. `Player` selects Start Game.
2. The system initiates the first level.
3. `Player` finishes the game.
4. The system prompts `Player` to type his/her name.
5. `Player` confirms.
6. The system stores a copy of the record permanently and sorts it according to the score made by the `Player`.
7. Options are being displayed to `Player`.

Alternative Flow of Events:

- `Player` ends up having 0 HP (game over message pops up and system continues from step 4)
- `Player` exits the game (system continues from step 4)

Use case 2:

Use case name: `PauseGame`

Participating actors: `Player`

Entry condition: `Player` is in game session.

Exit condition: `Player` continues to play the game.

Main Flow of Events:

1. `Player` presses pause button (ESC) during the game.
2. The system pauses the game and displays an intermediate options: **Continue, Settings, Restart, Exit.**
3. `Player` presses pause button (ESC) again.
4. The system continues the game.

Use case 3:

Use case name: `RestartGame`

Participating actors: `Player`

Entry condition: `Player` is in game session.

Exit condition: `Player` restarts the game.

Main Flow of Events:

1. `Player` presses ESC button during the game.
2. The system pauses the game and displays a menu that has options
    - Continue
    - Settings
    - Restart
    - Exit
3. `Player` selects Restart.
4. The system prompts user to answer: “Are you sure you want to restart the game?”
5. `Player` confirms.
6. System restarts the game.

Alternative Flow of Events:

- `Player` declines.
- Program continues to display the menu in paused condition.

Use case 4:

Use case name: `DisplayHighscores`

Participating actors: `Player`

Entry condition: Options are being displayed to `Player`.

Exit condition: Options are being displayed to `Player`.

Main Flow of Events:

1. `Player` selects Highscores while options are being shown.
2. The system reads the local repository and displays recorded scores in a decreasing order.
3. `Player` selects Return.
4. The system returns to the main menu.

Use case 5:

Use case name: `ChangeSettings`

Participating actors: `Player`

Entry condition: Options are being displayed to `Player`.

Exit condition: Options are being displayed to `Player`.

Main Flow of Events:

1. `Player` selects Settings option.
2. `Player` does necessary changes.
3. System applies changes to the game.

Alternative Flow of Events:

- `Player` exits menu without selecting option.

Use case 6:

Use case name: `DisplayCredits`

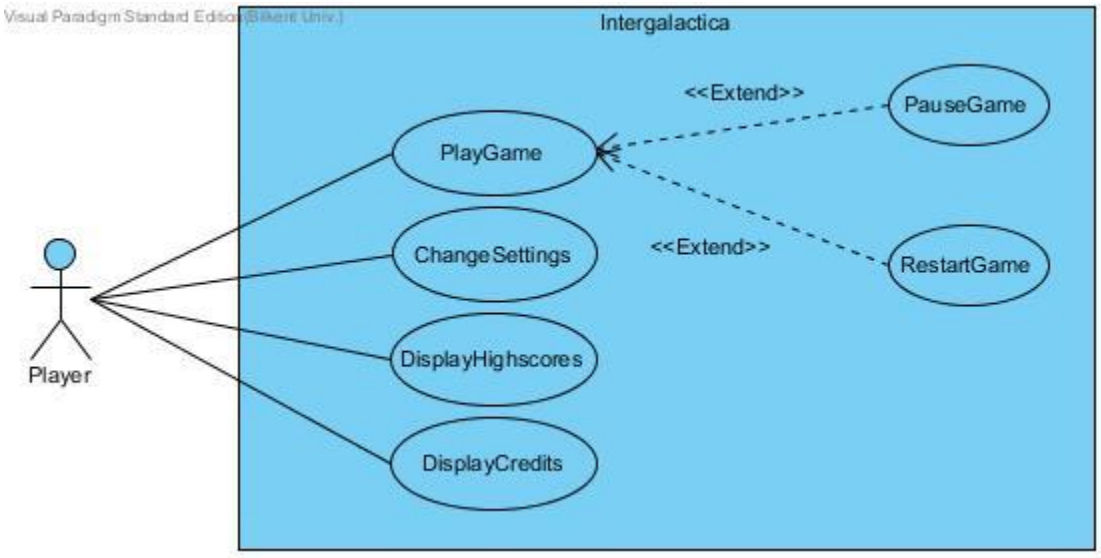
Participating actors: `Player`

Entry condition: Options are being displayed to `Player`.

Exit condition: Options are being displayed to `Player`.

Main Flow of Events:

1. `Player` selects Credits while Options are being displayed.
2. The system reads the local repository and displays recorded data.
3. `Player` selects Return.
4. The system displays Options back again.



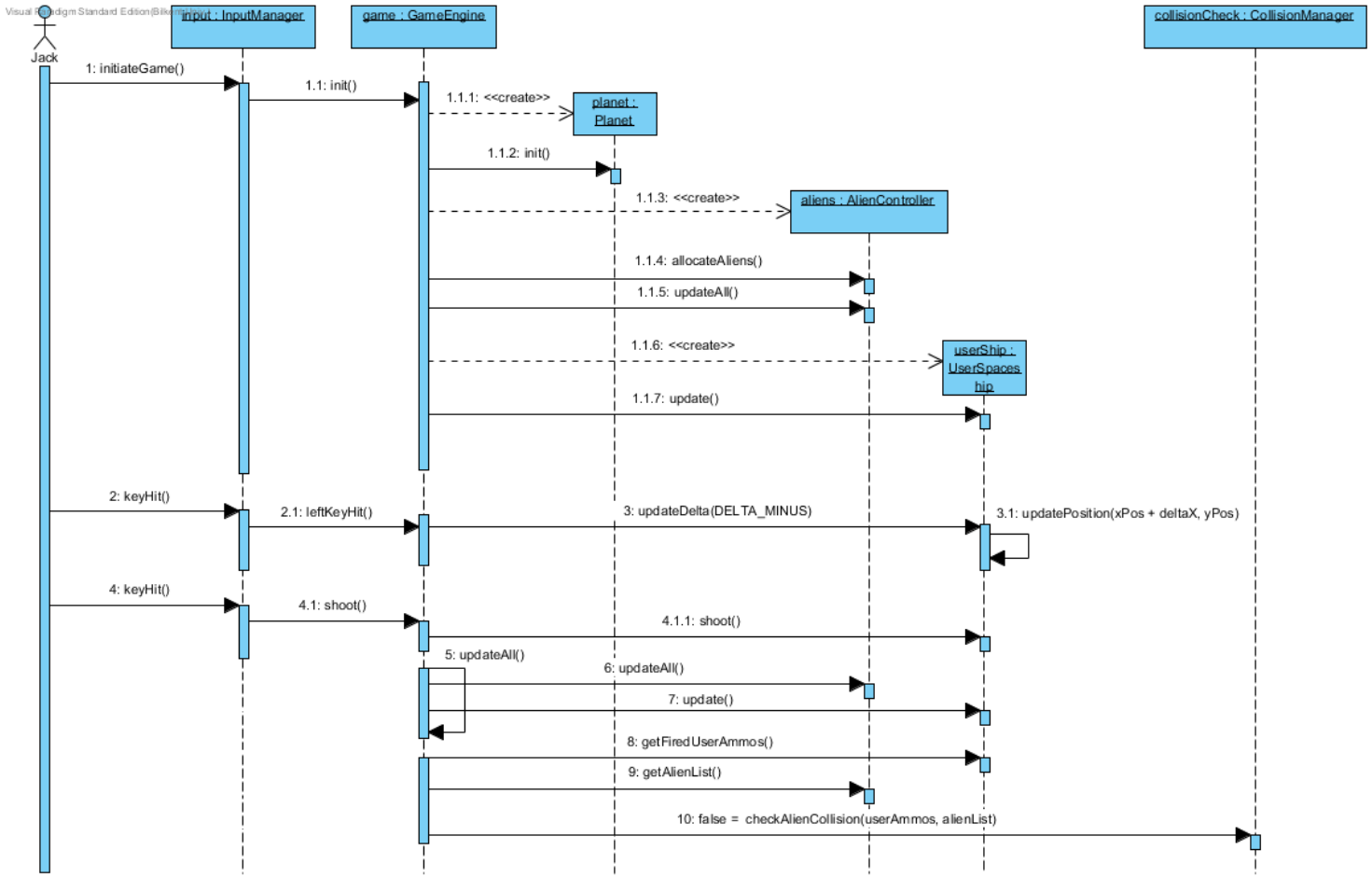
Use case Diagram for Intergalactica displaying functional view from user's perspective.

6. Dynamic Models

Scenario 1

Name: `Play Game`

Scenario: `Player Jack` is already in the main menu. He chooses start game option. When the game deploys he first moves left and fires a shot but misses it.



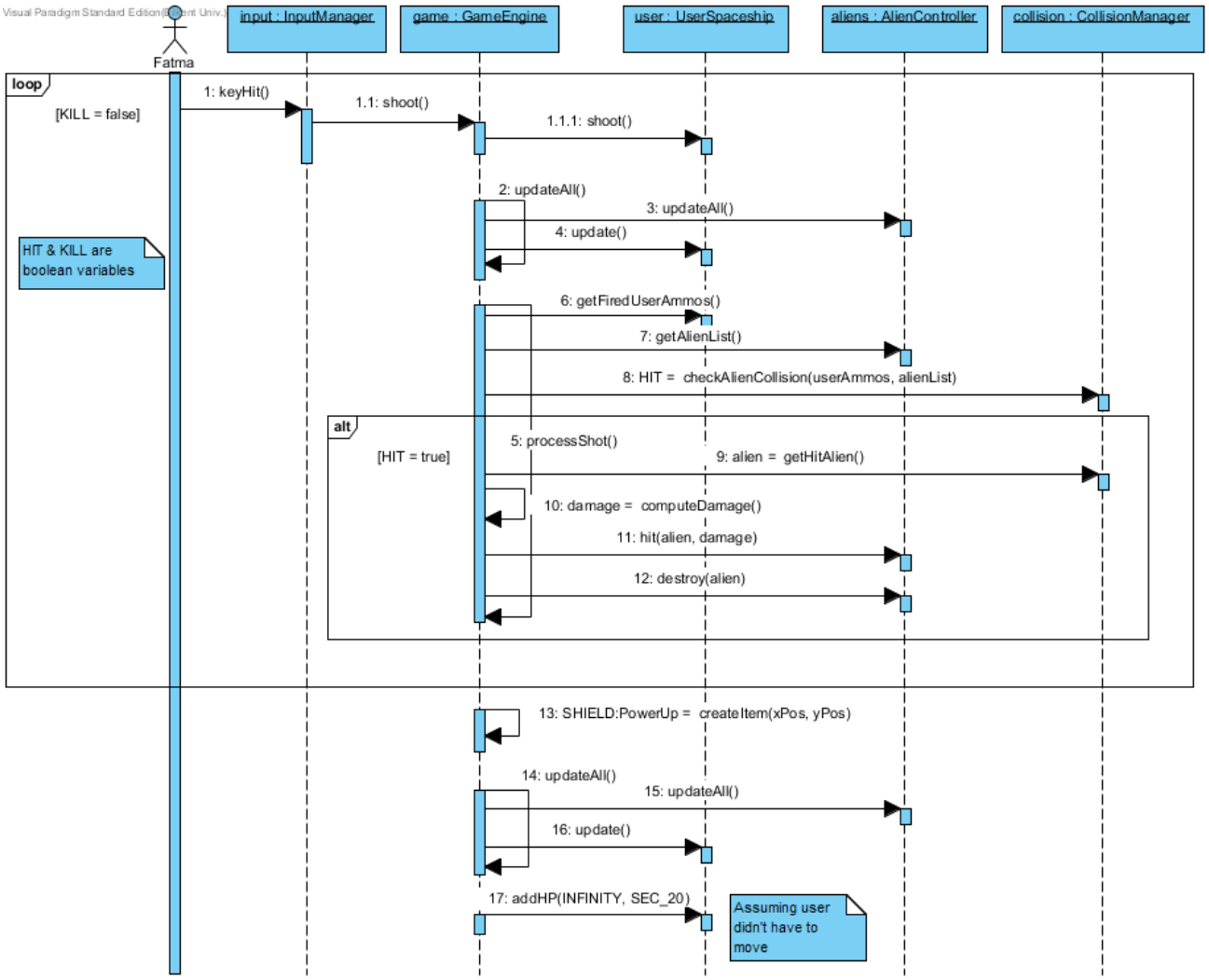
Sequence Diagram 1, depicting the chain of events occur when user successfully shoots but misses an `Alien`. This diagram also describes how killing mechanism works.

> ***Be Informed:*** KeyHit() by itself does not convey any information about which key was used, since this report is not Design, we did not designate KeyListener interface on class diagram which is implemented by InputManager. However InputManager actually uses KeyListener interface to understand which key was pressed.

Scenario 2

Name: `Item Drop`

Scenario: `Player Fatma` has already started playing the game and is currently at combat. She keeps shooting at a particular `Alien` until it gets destroyed. After the destruction, the alien drops an item and `Fatma` manages to collect it. When she collects, she realizes that she obtained a shield power up.

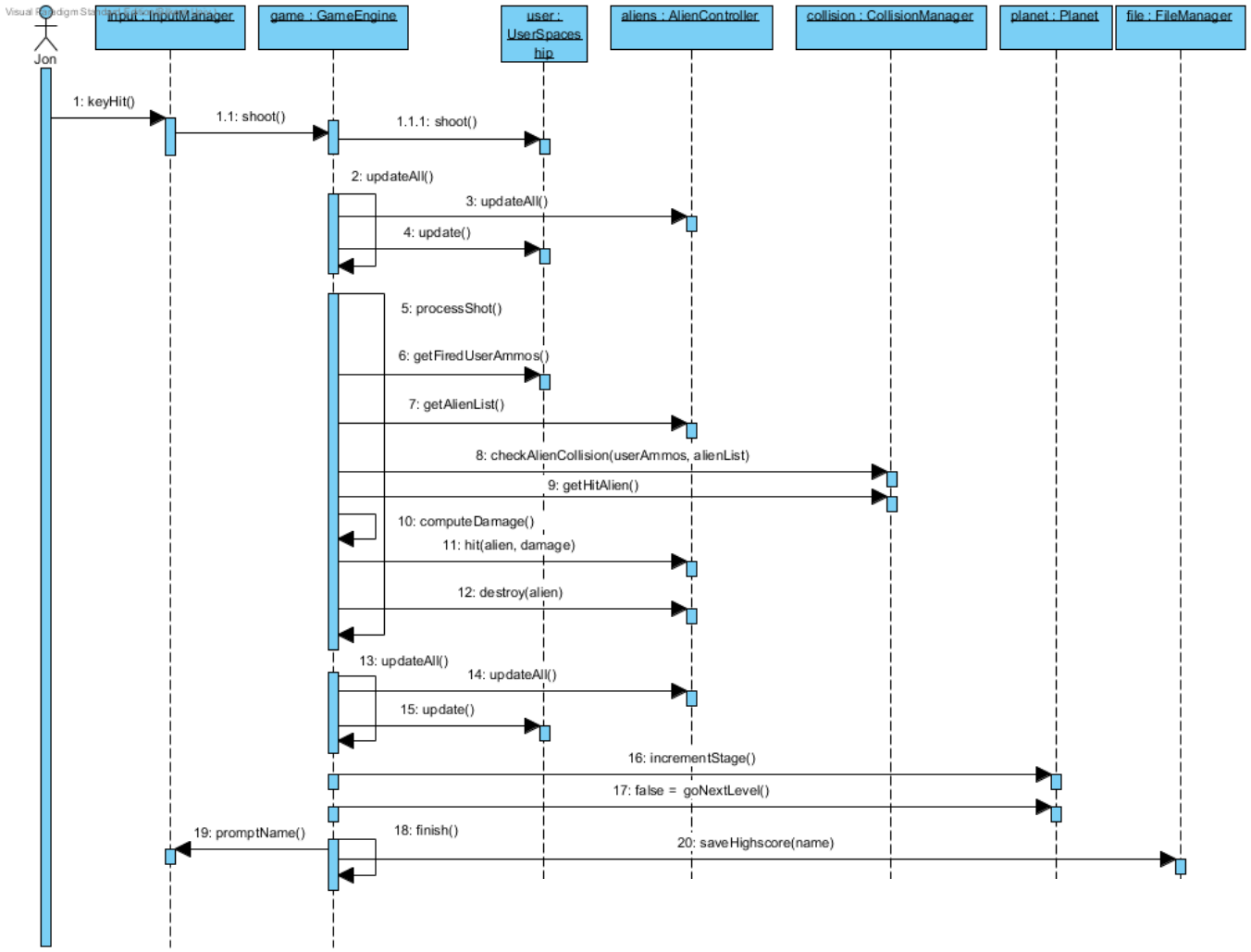


Sequence Diagram 2, depicting the chain of events occur when user successfully destroys an `Alien` and collects a Power UP.

Scenario 3

Name: `Finishing the Game`

Scenario: `Player Jon` has finished all the levels until the last one. He is now on the last stage of the last level. He kills the last remaining `Alien` and finishes the game. Saves his name and quits.

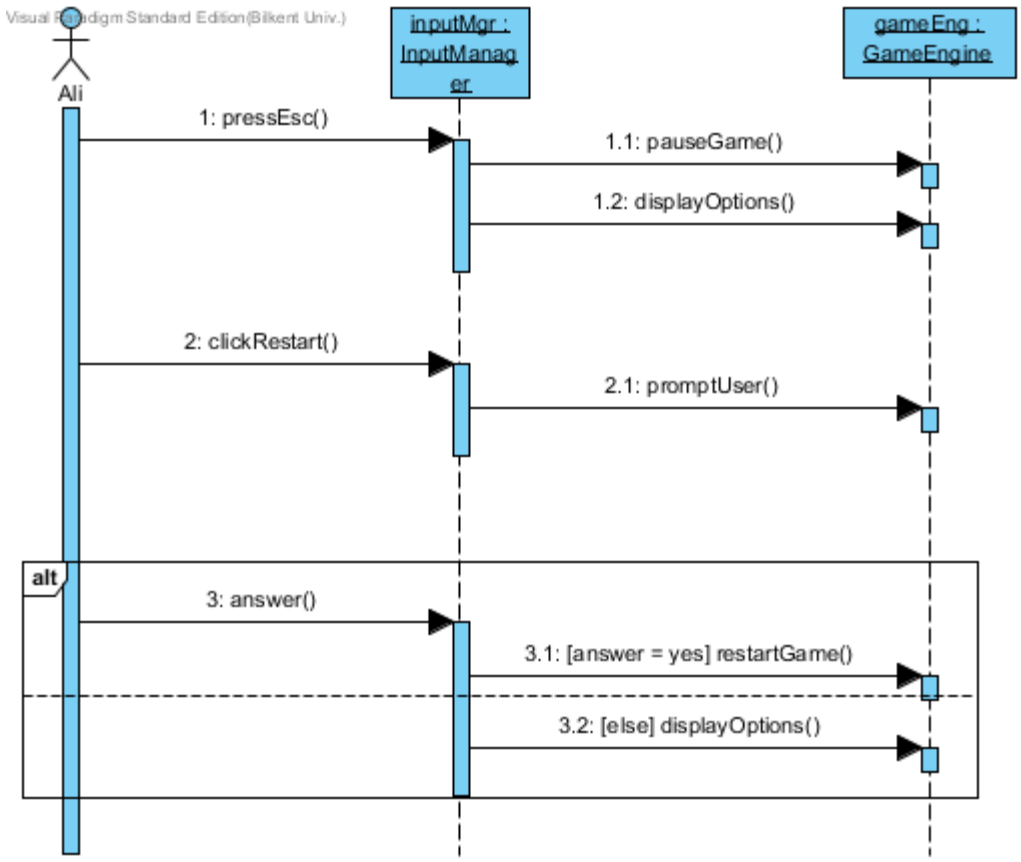


Sequence Diagram 3, depicting the chain of events occur when game reaches the end.

Scenario 4

Name: `Restarting the Game`

Scenario: `Player Ali` is playing the game, he presses escape button. The game pauses and displays options one of which is Restart. `Ali` chooses Restart and game prompts him for verification. If Ali chooses yes, the game restarts otherwise game carries on displaying options.

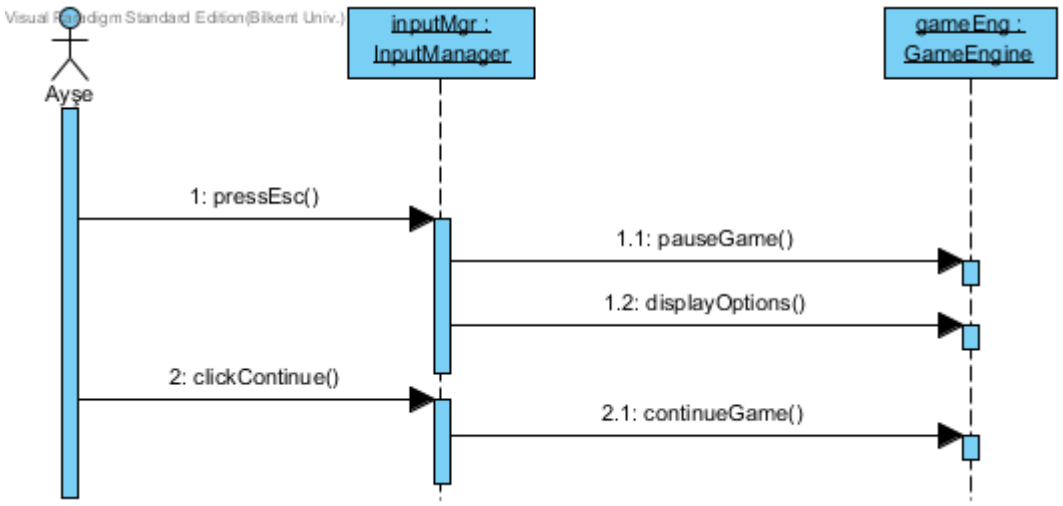


Sequence Diagram 4, depicting Restart functionality.

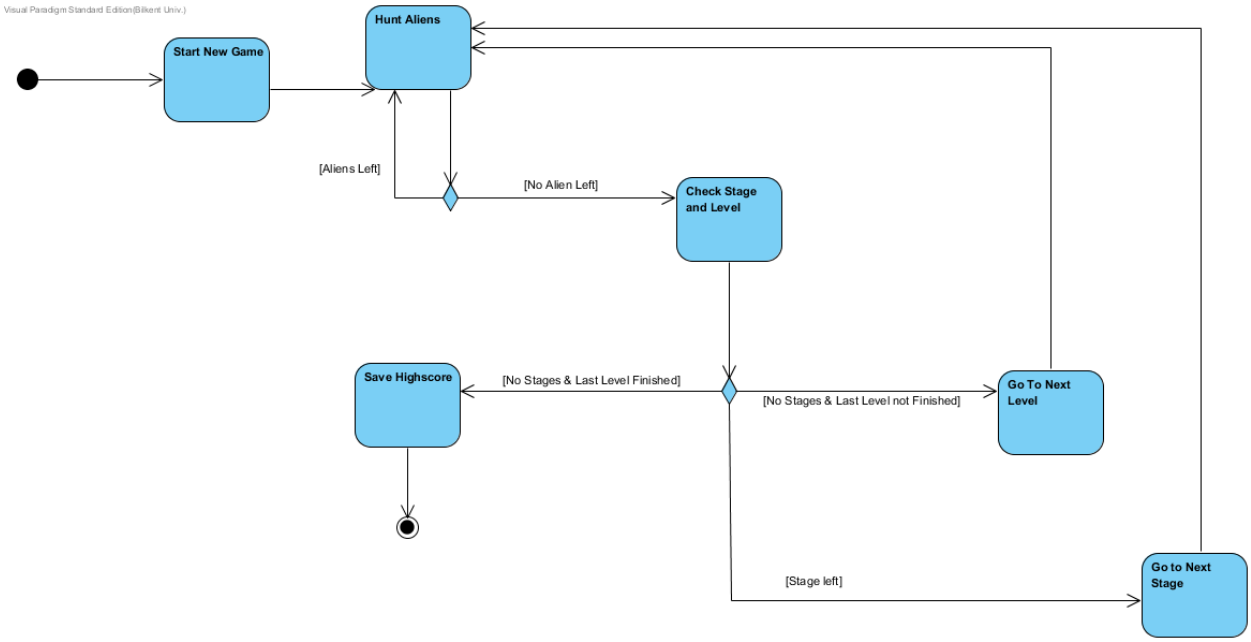
Scenario 5

Name: `Pausing the Game`

Scenario: `Player Ayşe` is playing the game, he presses escape button. The game pauses and displays options one of which is Continue. `Ayşe` chooses Continue and the game prompts continues.

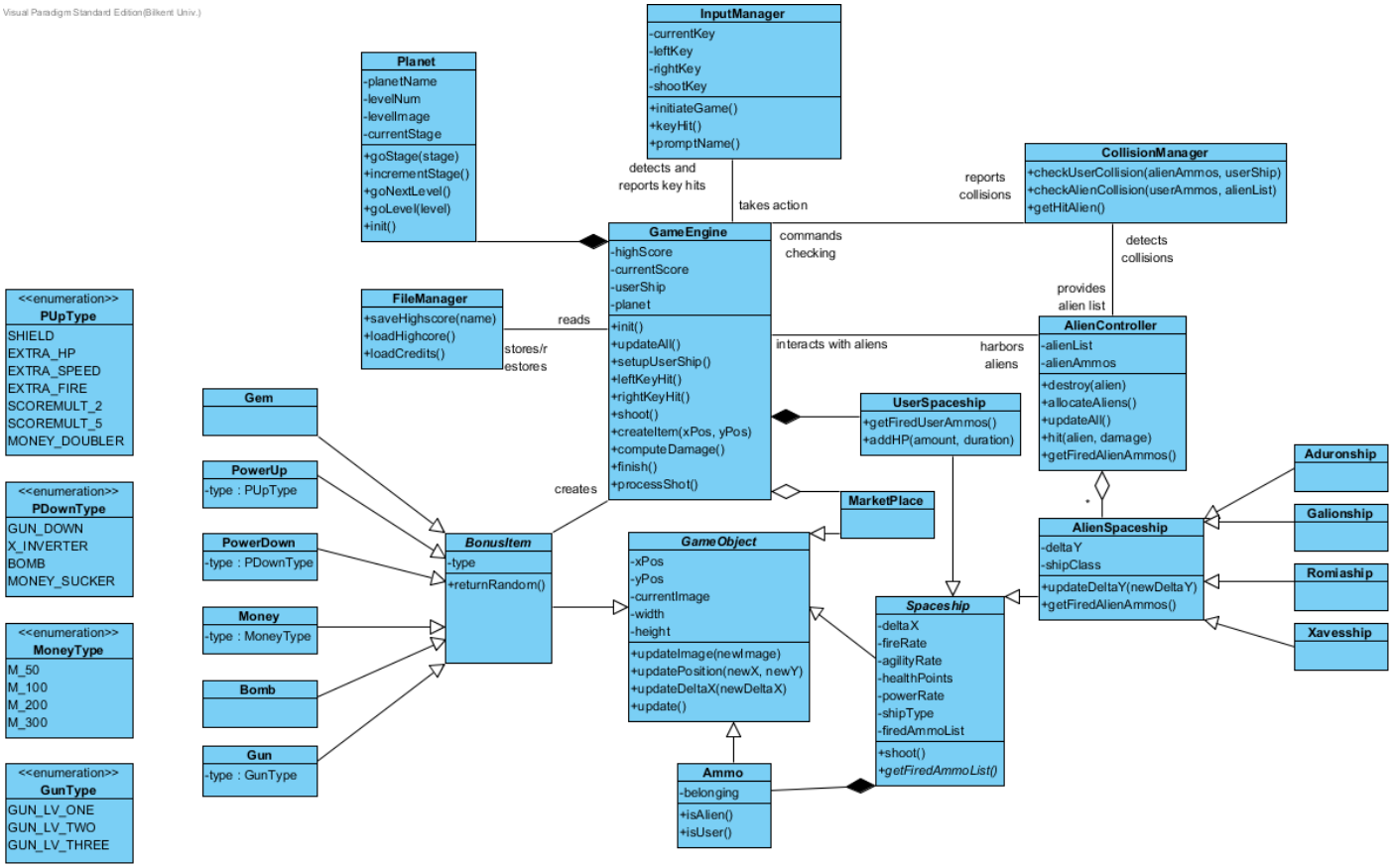


Sequence Diagram 5, depicting Pause functionality.



Activity Diagram for the overall logic of the game.

7. Object and Class Model



Be Informed: KeyHit() by itself does not convey any information about which key was used, since this report is not Design, we did not designate KeyListener interface on class diagram which is implemented by InputManager. However InputManager actually uses KeyListener interface to understand which key was pressed.

8. User Interface – Navigational Paths and Screen Mock-ups

Main Menu

When game runs, players will first see “Main Menu “. On this screen, there will the logo of InterGalactica, along with four buttons, “START GAME”, “HIGH SCORES” , “CREDITS” and “CHANGE SETTINGS”.

Play Game

When a player clicks on the button “START GAME” on the main menu, he will be directed to the game screen. In the Figure 5 which illustrates the overall game screen, spaceships are denoted by empty squares. The background of game screen and the spaceships will change from level to level. The player will be able to see the highest score achieved so far and the score that he currently has, on the top of the game screen. Furthermore, the player will be able to see his remaining “health points” by the bar located near his spaceship.

In-Game Menu

A player will access the “In-Game Menu” if he pauses the game by ESC key of his keyboard. In “In-Game Menu”, there will be several buttons to control the game. The player will be able to resume the game by clicking on the “RESUME” button. He will be able to go back to the “Main Menu” by clicking on the “HOME” button. Furthermore, he will be able to restart the game, without going back to the “Main Menu”, by clicking on the “RESTART” button. Player can also view help screen to get information about gameplay of InterGalactica by clicking on the help button(“?” button).

View Help

On the “In-Game Menu” there will be a help button notated with a question mark. If a player clicks on that button, he/she will be able to view the instructions on how to play the game from “View Help” screen. There will be several images that illustrate the gameplay and written descriptions(instructions). After the player understands the gameplay, he will be able able to go back to the “In-Game Menu” by clicking on the “BACK” button.

Market Place

After the player finished each level, Market Place screen will appear automatically. In market place screen there will be images, prices and descriptions of available guns, spaceships and power-ups that the player can purchase with the money he collected through previous levels. Player's total money will be shown on the upper right corner of the screen. Player will be able to

Note that: Our Mockups are in a very primitive shape, because we'd like to try and add new styles that we are yet to decide, so we will present them now in their most tentative form.

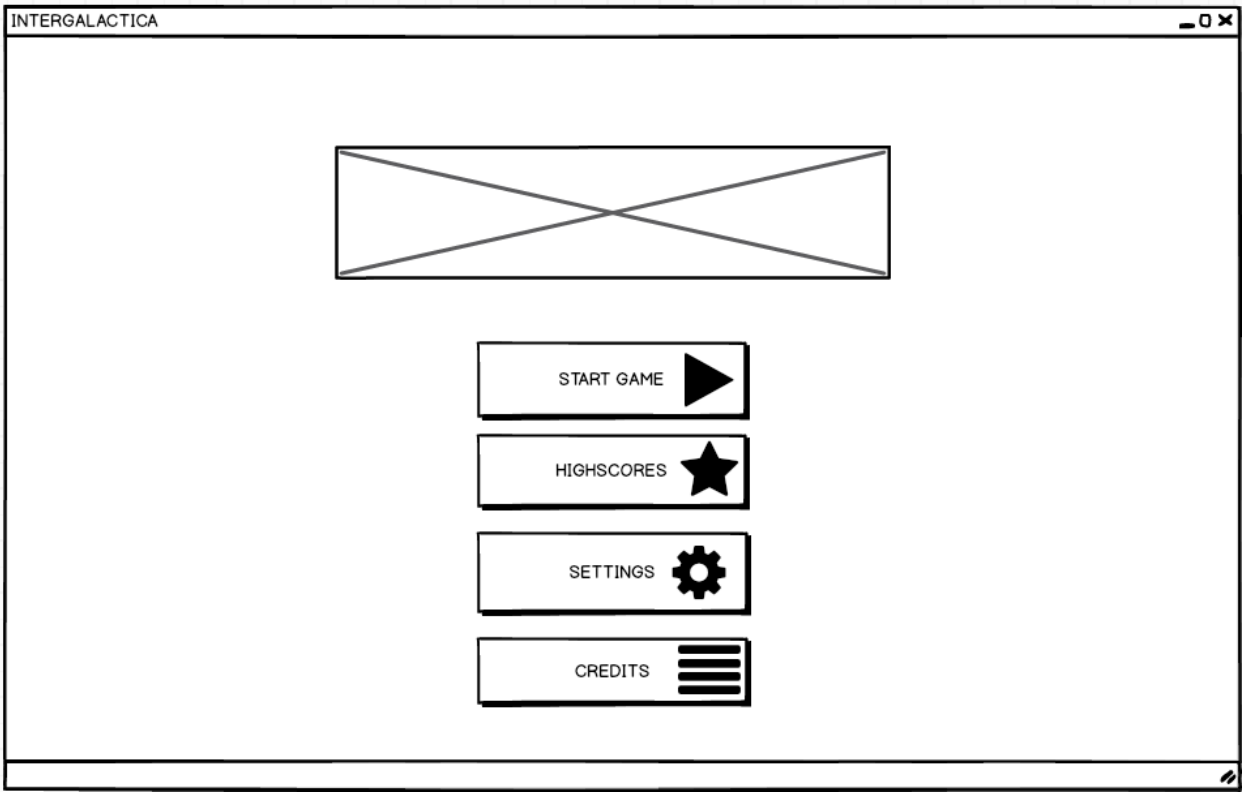


Fig. 1 Main Menu

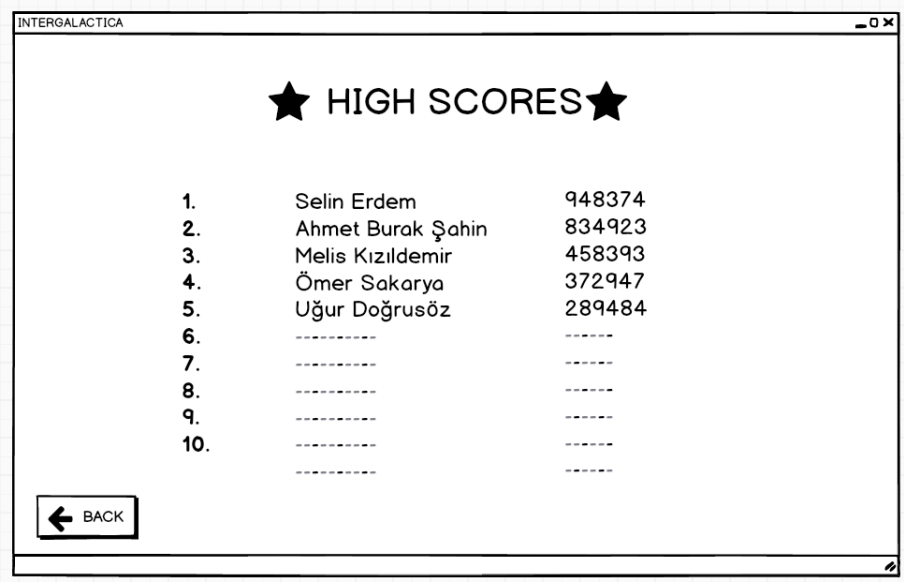


Fig. 2 Highscores

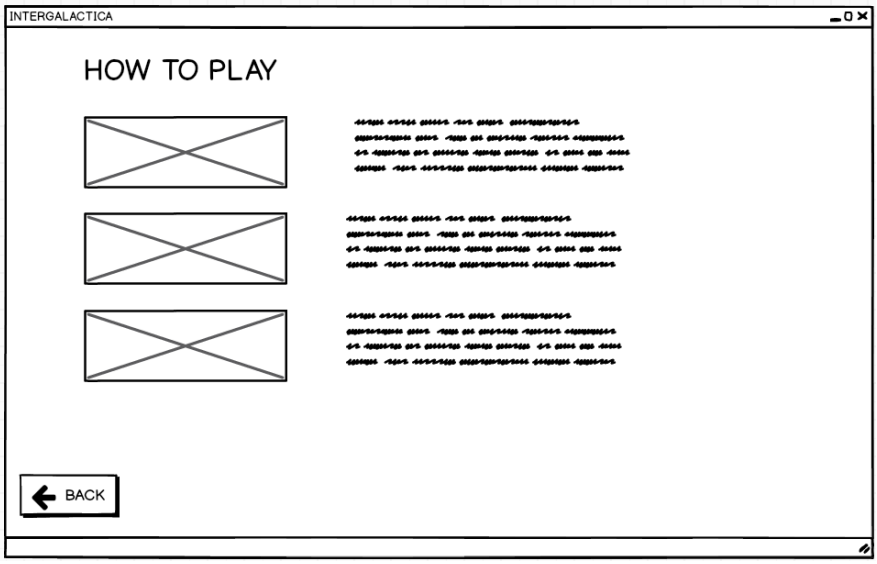


Fig. 3 Help Panel.

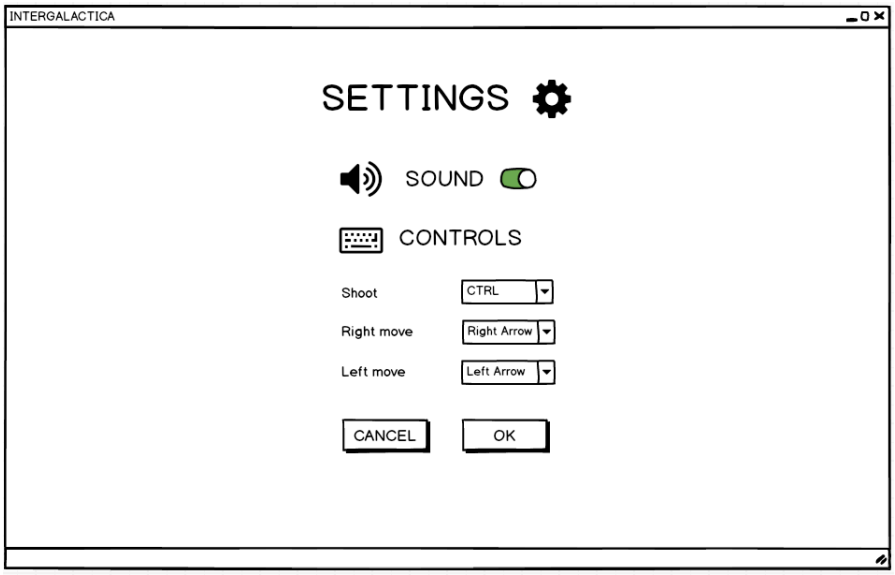


Fig. 4 Settings.

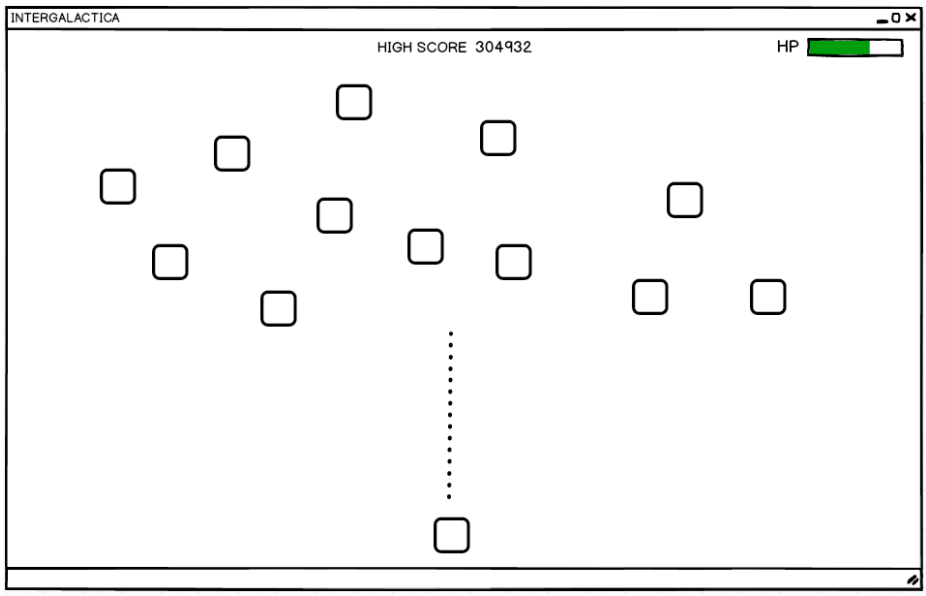


Fig. 5 In-Game view.

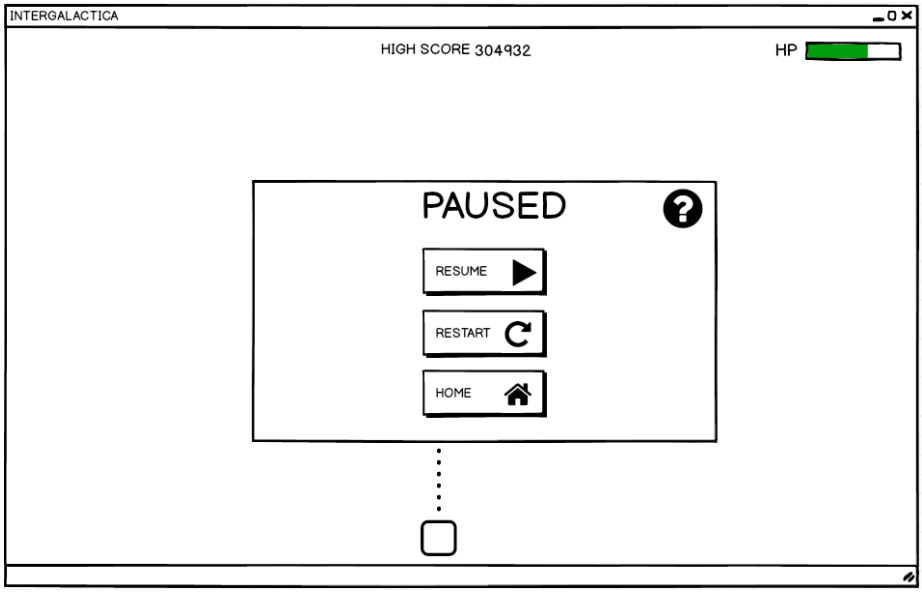


Fig. 6 Pause Menu.

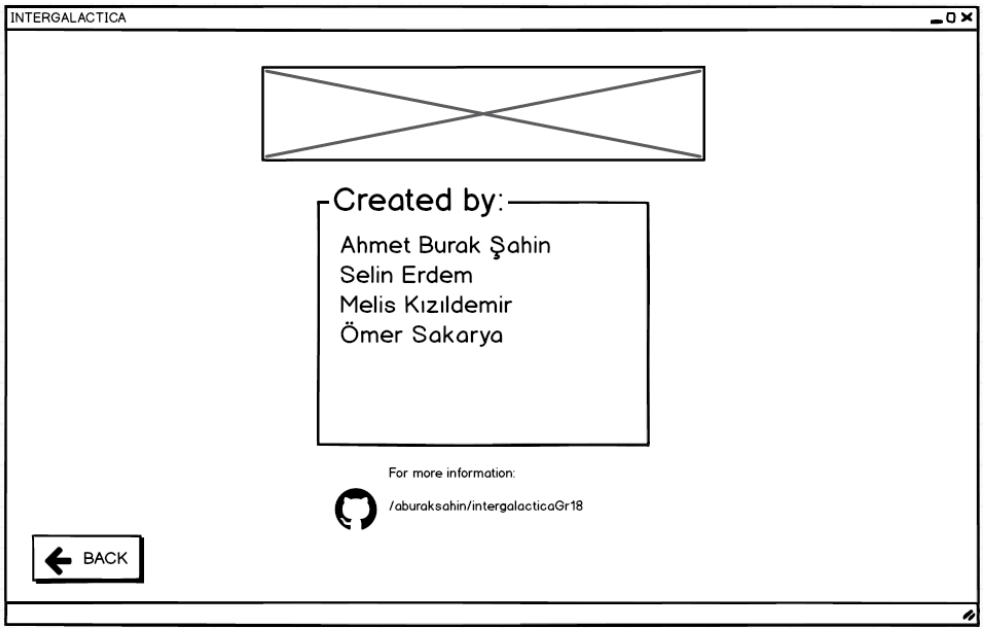


Fig. 7 Credits Panel.

9. References

[1] “Galaga.” Wikipedia. Wikimedia Foundation, n.d. Web. 11 Oct. 2016. (https://en.wikipedia.org/wiki/Galaga)

[2] “Free Sprites.” MillionthVector:. N.p., n.d. Web. 11 Oct. 2016. (http://millionthvector.blogspot.com.tr/p/free-sprites.html)

[3] “Ship Designs – 3D/3D graphic.” Darke:. N.p., n.d. Web. 11 Oct. 2016.

(http://stardrivegame.com/forum/viewtopic.php?f=1&t=8938)