Bilkent University

Department of Computer Engineering

Object Oriented Software Engineering Project

Project Short-Name: Space Despot

Analysis Report

Cihangir Mercan, Çağdaş Han Yılmaz, Fırat Sivrikaya, Gökçe Şakir Özyurt

Supervisor: Prof. Dr. Uğur Doğrusöz

Progress Report
Oct 29, 2016

Contents

Analysis Report

Project short-name: Space Despot

1 Introduction

Space Despot is an arcade game in which player controls a spaceship that moves upwards in a space through different levels. In the space, where player can move horizontally and vertically, there will be various space objects that try to destroy spaceship by attacking to it. Player's goal will be making his way up to the top by avoiding these space objects or shooting them similar to the game that we are influenced [1].

2 Overview

Player will start game by choosing between three types of spaceships, which are Fighter, Panzer and Lightning (Section 2.1). Game starts from level 1. There will be three levels. With each passing level, the vertical space length will increase and the space objects (Section 2.2) will become harder to avoid. If player shoots these space objects, they might drop power-ups or coins. Power-ups (Section 2.3) will make spaceship more powerful by giving it various specialties. Coins will help player to do upgrades (Section 2.4) for the spaceship at the end of each level. These upgrades improve spaceship's attributes.

Player's score will depend on how many levels passed, how many objects destroyed, how many power-ups collected and how many upgrades bought. If player passes all three levels or loses all of spaceship's health points (HP), game will be over. Then, player will be asked to give a name for the score. This score can be shown at the high scores.

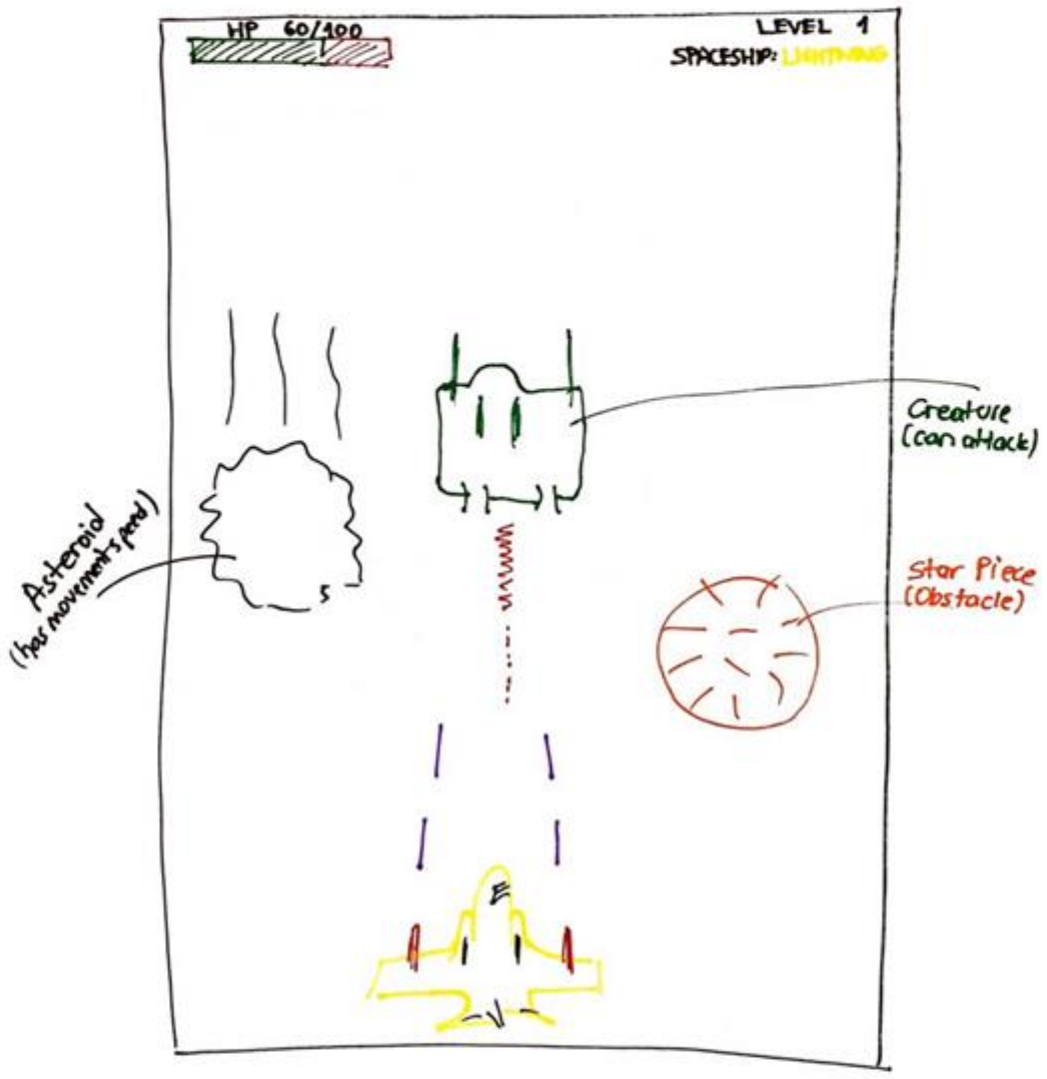


Figure 1 - Overview

2.1 Spaceships

2.1.1 Fighter

Fighter spaceship has the advantage in attack damage as it has the most attack damage compared to other spaceships. Its movement speed is medium but its HP is low. Fighter is advised for beginner players.

Base Stats

- HP: 75
- Movement Speed: 100
- Attack Damage: 20

2.1.2 Panzer

Panzer spaceship can withstand more damage due to having more HP than the other two spaceships. However, its heaviness makes it less volatile so it will have the lowest speed. Panzer deals medium damage. Panzer is advised to intermediate and advanced players.

Base Stats

- HP: 150
- Movement Speed: 50
- Attack Damage: 15

2.1.3 Lightning

Lightning spaceship is the most advantageous spaceship in terms of its movement speed. Its swiftness enables player to deal with creatures and obstacles more easily. However, it is not advised to players who prefer to get in combat with mobs frequently, since it has the lowest attack damage compared to other spaceships. Also, it has medium HP. Lightning is advised to intermediate and advanced players.

Base Stats

- HP: 100

- Movement Speed: 150
- Attack Damage: 10

2.2 Space Objects

2.2.1 Mobs

Mobs will have their own HP which makes them vulnerable to the user attack. They will also have a unique movement speed in addition to the background speed. So, they will try to damage the spaceship by shooting or crushing into it. There are three types of mobs: creatures, asteroids and bosses.

- Creatures

Creatures are simple objects that try to deal damage to spaceship by shooting.

<u>Stats</u>

- HP: 40
- Movement Speed: 100
- Attack Damage: 20

- Asteroids

Asteroids are objects that appear from different directions. They will try to crush into spaceship rapidly and do damage.

<u>Stats</u>

- HP: 20
- Movement Speed: 200
- Attack Damage: 40

- Bosses

At the end of each level, player will face bosses in order to finish level. They will have better attributes than the other mobs and will require player to try harder.

<u>Stats of Boss #1</u>

- HP: 300
- Movement Speed: 300
- Attack Damage: 30

<u>Stats of Boss #2</u>

- HP: 450
- Movement Speed: 250
- Attack Damage: 60

<u>Stats of Boss #3</u>

- HP: 600
- Movement Speed: 200
- Attack Damage: 90

2.2.2 Obstacles

Obstacles don’t have HP, which means they cannot be destroyed by the player. They don’t have movement speed, they move as the game background moves. Also, they don’t have attack abilities. When in contact with spaceship, the spaceship gets destroyed and the game ends for the player. Obstacles can be dodged by moving the spaceship to the appropriate positions in the map. There are two types of obstacles: stars and black holes.

- Stars

Stars are obstacles that cause spaceship to get destroyed immediately when in contact. There is more than one type of stars but they only differ in colors.

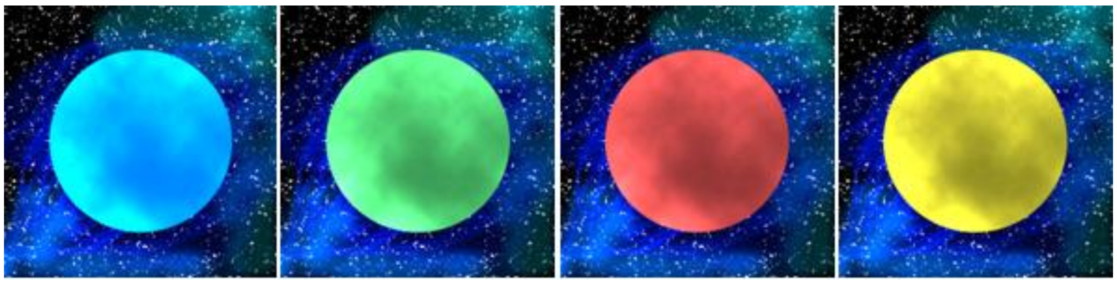


Figure 2 - Stars can have different colors

- Black Holes

Black holes are obstacles that devour the spaceship when in contact. Since their size is quite larger than the stars, they are harder to avoid. Hence, they occur more rarely than the stars.

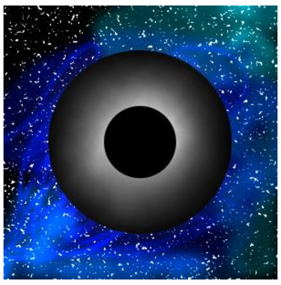


Figure 3 - Black Hole

2.3 Power-Ups

Power-ups drop when mobs are destroyed. Player can only hold one power-up at a time. If player collects new power-up, it will overwrite the previous one. Player will be able to activate these power-ups by pressing Space. There are four types of power-ups.

2.3.1 Invulnerability

It will make spaceship invulnerable for 3 seconds when it is activated.

2.3.2 Laser Gun

It will generate a laser ray that instantly kills mobs along the line. If it hits boss, it will make three times more damage than the base attack damage of spaceship.

2.3.3 Hyper Drive

It will increase the spaceship’s speed by %100 for five seconds.

2.3.4 Repair

It will recover the spaceship’s HP by %25 of its max HP.

2.4 Upgrades

Upgrades can be done at the end of each level with coins. Coins drop when mobs are destroyed or they randomly appear at the space with varying amounts. Upgrades cost amount of $1000 * (2^{(n-1)})$ where n is how many times it is done before.

2.4.1 Maximum Health

It will increase spaceship’s max HP by %5. It can be bought maximum of 20 times.

2.4.2 Attack Damage

It will increase spaceship’s attack damage by %5. It can be bought maximum of 20 times.

2.4.3 Movement Speed

It will increase spaceship’s base movement speed by %5. It can be bought maximum of 20 times.

3 Functional Requirements

- Player can move horizontally and vertically with arrow keys.
- Player can shoot by pressing X.
- Player can use power-ups by pressing Space.
- Player can pause the game.
- Player can choose level to begin with.
- Player can set key for shooting and using power-ups from the settings.
- Player can enable/disable sound from the settings.
- Player can access help menu which contains information about gameplay and controls.
- Player can see top ten high scores.

4 Non-functional Requirements

- Game will be designed and implemented by applying the principles that we learn during class. We will try to create most possible optimized game. We are aiming to decrease the input lag as much as possible in order to enhance gameplay experience.
- Game will have a good-looking and attractive user interface. Besides, we will work a lot for the smoothness.
- Game will be suitable for extensions. It will be easy to add new features and functionalities to the game in the future.

5 Pseudo-functional Requirements

- Game will be implemented using Java.
- Game images will be designed using Adobe Photoshop.

6 System Models

6.1 Use-Case Model

6.1.1 Use-Case Scenarios

Use-Case 1

Play Game

Primary Actor: Player

Main Success Scenario

1. Player starts the game by choosing spaceship type.

2. Game starts from level 1.

3. After three seconds countdown, spaceship is ready to move.

4. Player passes all levels.

5. The score is at the top ten scores.

6. The system asks for a name for the score gained.

6. Player enters the name.

7. The score is at placed to high scores.

8. Player quits the game.

Alternate Flow

4a. Player loses all health points at any level.

 4a1. The system displays game-over message.

 4a2. The system redirects player to step 5.

5a. The score is not at the top ten scores.

 5a1. The system redirects player to step 8.

6a. Player does not enter the name and directly quits the game.

Use-Case 2

Pause Game

Primary Actor: Player

Main Success Scenario

1. Player presses ESC.
2. The system displays pause menu.
3. Player continues the game.
4. The game continues.

Alternate Flow

3a. Player change settings during pause.

  3a1. The system redirects player to the settings menu.

3b. Player quits the game.

  3b1. If the score is at the top ten scores, player enters a name for the score.

Use-Case 3

Continue Next Level

Primary Actor: Player

Main Success Scenario

1. Player successfully passes one of levels.
2. The system displays menu for next level.
3. Player does upgrades for the spaceship with coins.
4. Player continues with next level.

Alternate Flow

3a. Player does not do upgrades and save coins.

    3a1. Player goes to step 4.

4a. Player quits the game.

    4a1. If the score is at the top ten scores, player enters a name for the score.

Use-Case 4

View Help

Primary Actor: Player

Main Success Scenario

1. Player chooses to view help.
2. The system displays information about gameplay and controls.
3. Player quits viewing help.

Use-Case 5

Change Settings

Primary Actor: Player

Main Success Scenario

1. Player chooses to view settings.
2. Player sets new key for shooting and disable sound.
3. Player quits viewing settings.

Alternate Flow

2a. Player does not change anything in settings.

    2a1. Player goes to step 3.

Use-Case 6

View High Scores

Primary Actor: Player

Main Success Scenario

1. Player chooses to view high scores.
2. The system displays top ten high scores.
3. Player quits viewing high scores.

6.1.2 Use-Case Diagram

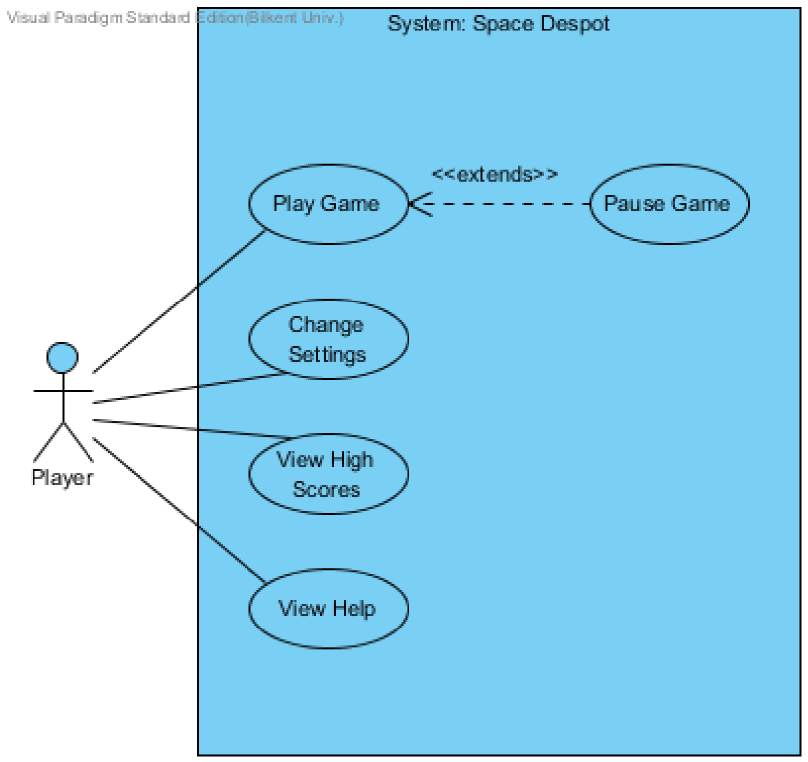


Figure 4 - Use-Case Diagram

6.2 Dynamic Models

In this section, we will explain some important scenarios about Space Despot. First three scenarios will be about game play:

- Execute Game: choose spaceship type and execute game.

- Use Power-up: shoot, destroy creature, pick power-up, use power-up.

- Do Upgrade: do upgrade at the end of level and start next level.

Then, the last one will be about menu:

- Menu: view settings, change settings, view high scores, view help.

6.2.1 Sequence Diagram

#1

Play Game

Primary Actor: Ali

Scenario

Ali double clicks the game icon in the desktop. Then game opens, he chooses to start game. He chooses spaceship type Fighter. Then, he starts game. Level 1 opens. He sees his spaceship. After three seconds countdown, game starts. Space mobs and space obstacles start to appear, they increase in count as time passes. Some of them also shoot bullets periodically.

Description

After Ali starts game, startGame(LEVEL_1, Fighter) is called by Game and GameEngine is ready after that point. It draws spaceship and hp bar. Then, it makes requests to Mob Sender Services (there are 3 of them), Obstacle Sender Services(there are 2 of them), SpaceMobShootingController, UpdateController. MobSenderServices and ObstacleSenderServices add space mobs and obstacles to the spaceMobsInSpace and spaceObstaclesInSpace periodically with regard to level. SpaceMobShooting services make space mobs shooting bullets. These bullets are added to bulletsInSpace as CreatureBullet or BossBullet according to mob’s type. UpdateController is the heart of everything. It gets help from CollisionCheckerService which checks and handles collisions. UpdateController also calls drawObjects() and drawHPbars() from GameEngine in each iteration for repainting. It also updates score.

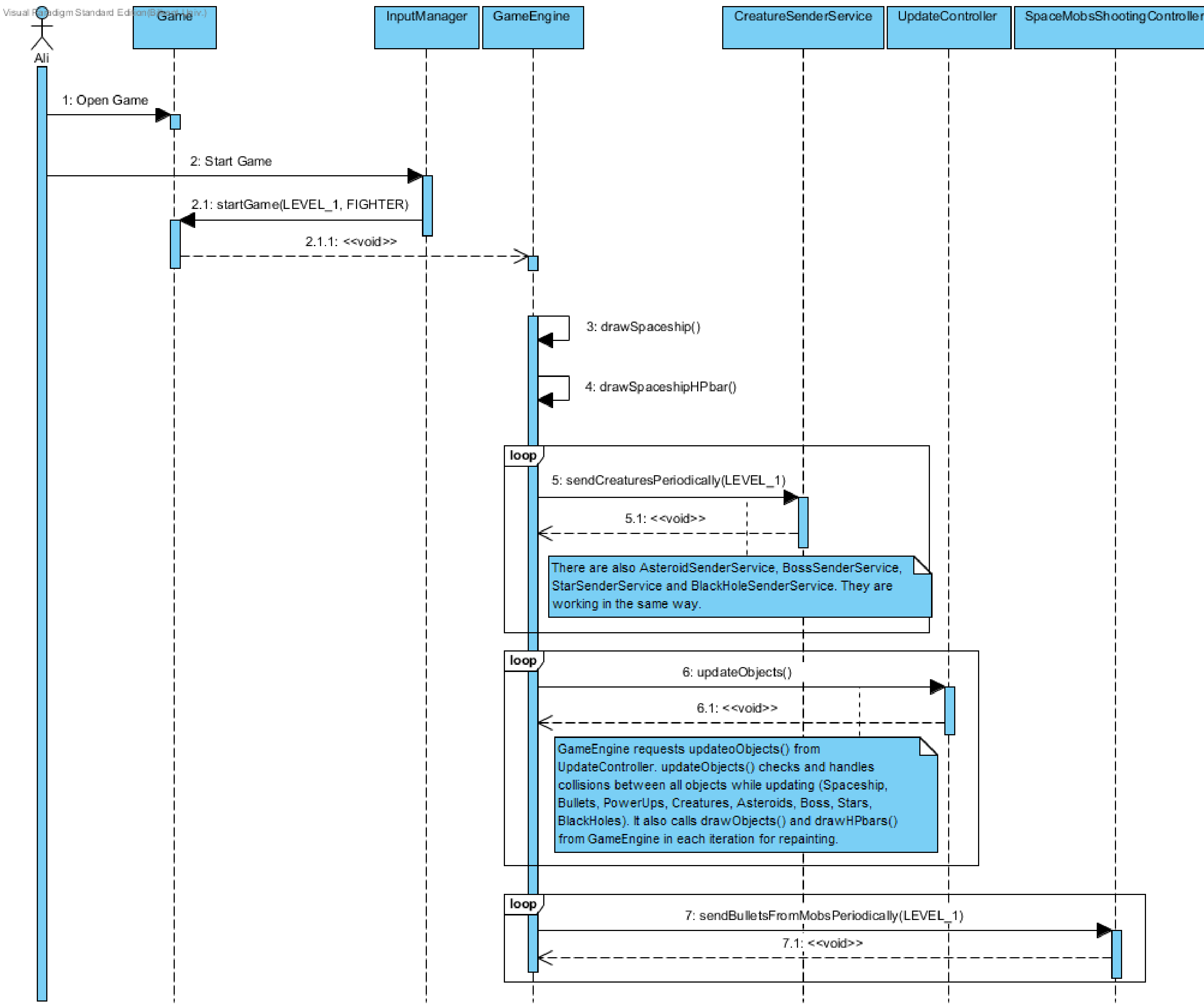

Visual Paradigm Standard Edition(Bilkent Univ.)
Ali
Game
InputManager
GameEngine
CreatureSenderService
UpdateController
SpaceMobsShootingController
1: Open Game
2: Start Game
2.1: startGame(LEVEL_1, FIGHTER)
2.1.1: <<void>>
3: drawSpaceship()
4: drawSpaceshipHPbar()
loop
5: sendCreaturesPeriodically(LEVEL_1)
5.1: <<void>>
There are also AsteroidSenderService, BossSenderService, StarSenderService and BlackHoleSenderService. They are working in the same way.
loop
6: updateObjects()
6.1: <<void>>
GameEngine requests updateoObjects() from UpdateController. updateObjects() checks and handles collisions between all objects while updating (Spaceship, Bullets, PowerUps, Creatures, Asteroids, Boss, Stars, BlackHoles). It also calls drawObjects() and drawHPbars() from GameEngine in each iteration for repainting.
loop
7: sendBulletsFromMobsPeriodically(LEVEL_1)
7.1: <<void>>

Figure 5 – Play Game Sequence Diagram

#2

Use Power-up

Primary Actor: Ali

Scenario

Ali uses arrow keys and moves his spaceship. He shoots and destroys one of creatures. Creature drops power-up that increases movement speed (HYPER_DRIVE) luckily. Ali press Space and uses it. Hence, spaceship's movement speed increases for five seconds.

Description

While UpdateController is doing its work: send mobs, make mobs shoot, redraw objects etc., Ali uses arrow keys and moves his spaceship. Then, he shoots. InputManager handles these. When he uses arrow keys, spaceship's move() method is called and when he shoots, spaceship's shoot() method returns SpaceshipBullet. This bullet drops to the bulletsInSpace in GameEngine. Then this bullet hits the creature, creature gets destroyed. Since its hasPowerUp returns true, its dropPowerUp() is called. Then, this method returns PowerUp to the SpaceItemsInSpace. Next, Ali's spaceship collides with this power-up. Because currentPowerUp is null, Ali is able to collect power-up. pickPowerUp(PowerUp) is called and currentPowerUp becomes PowerUp. Then, Ali presses Space bar, usePowerUp() uses currentPowerUp and spaceship's velocityX and velocity increases for 5 seconds.

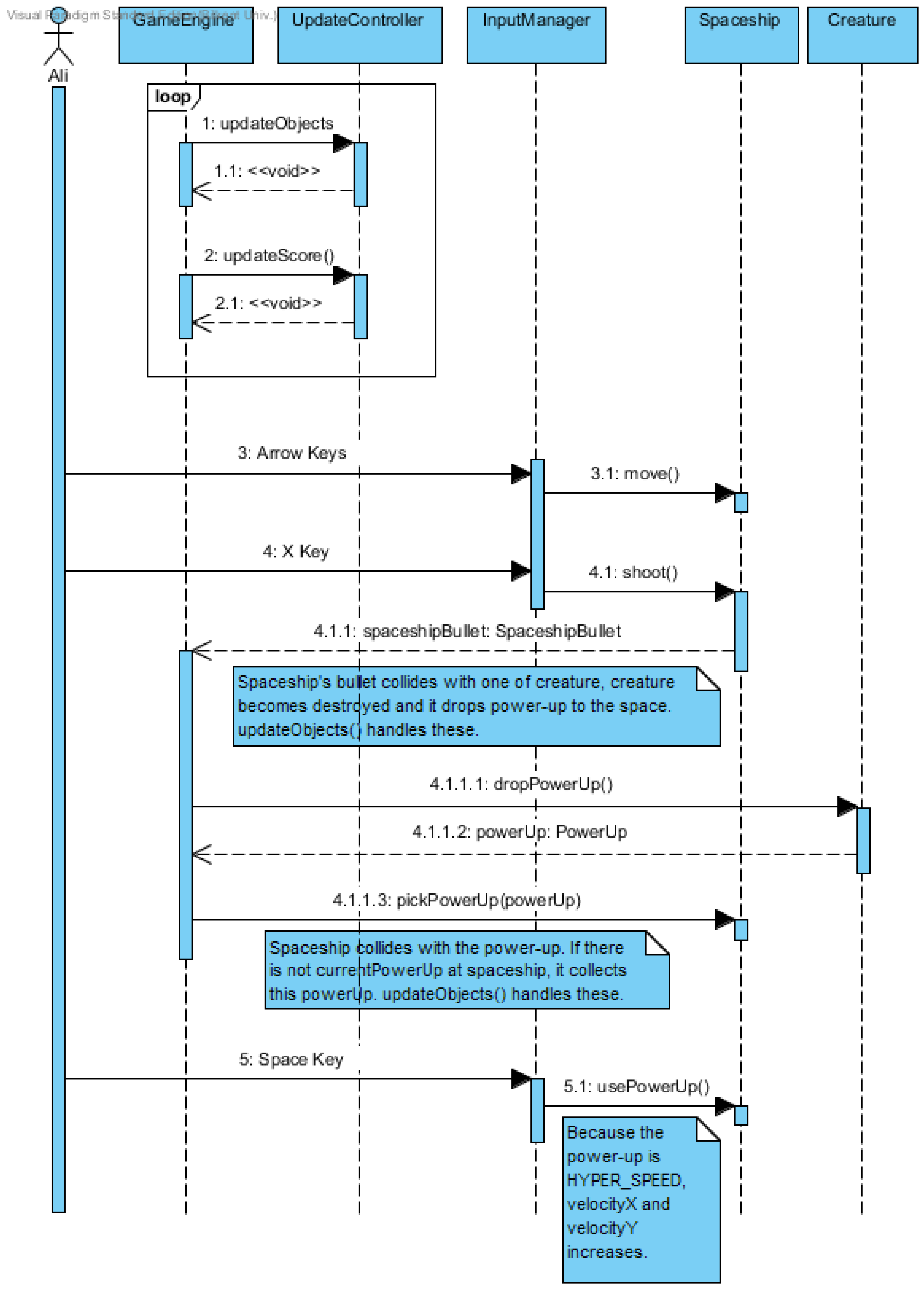

Visual Paradigm Standard Edition(Bilkent Univ.)
Ali
GameEngine
UpdateController
InputManager
Spaceship
Creature
loop
1: updateObjects
1.1: <<void>>
2: updateScore()
2.1: <<void>>
3: Arrow Keys
3.1: move()
4: X Key
4.1: shoot()
4.1.1: spaceshipBullet: SpaceshipBullet
Spaceship's bullet collides with one of creature, creature becomes destroyed and it drops power-up to the space. updateObjects() handles these.
4.1.1.1: dropPowerUp()
4.1.1.2: powerUp: PowerUp
4.1.1.3: pickPowerUp(powerUp)
Spaceship collides with the power-up. If there is not currentPowerUp at spaceship, it collects this powerUp. updateObjects() handles these.
5: Space Key
5.1: usePowerUp()
Because the power-up is HYPER_SPEED, velocityX and velocityY increases.

Figure 6 – Use Power-up Sequence Diagram

#3

Do Upgrade

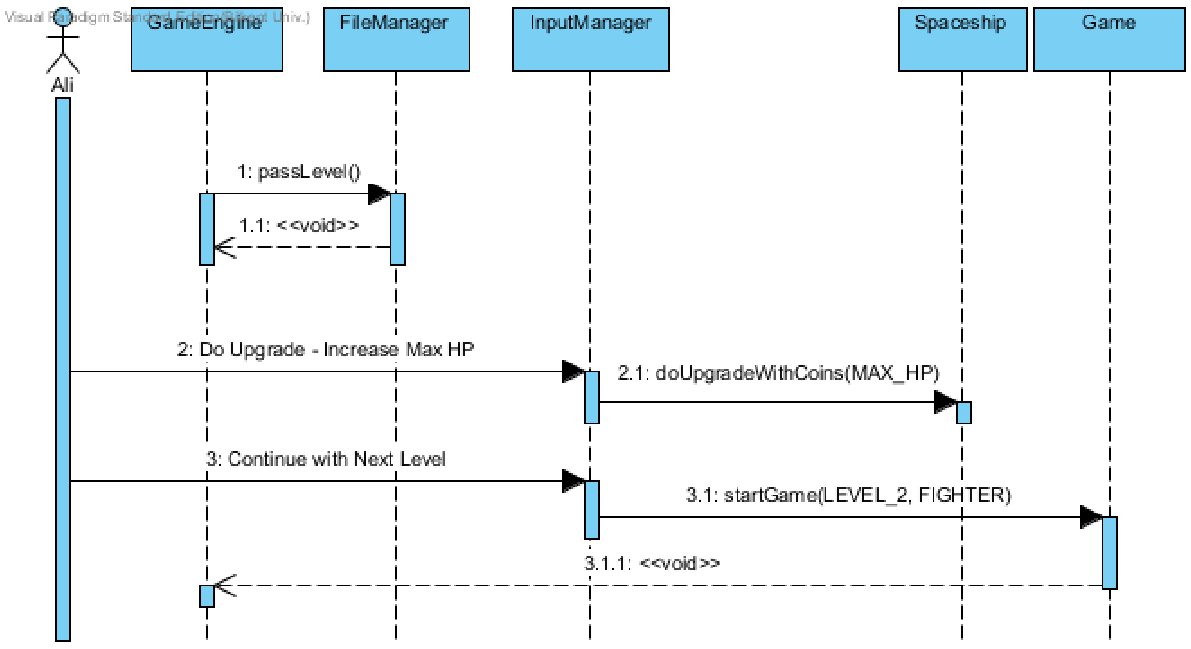
Primary Actor: Ali

Scenario

Coins drop from space mobs like the way power-ups drop. Ali has collected 1700 coin at level 1. He successfully passes level by not dropping spaceship's HP to zero. Then, Ali comes to the space station, where Ali is able to do upgrades with the coins collected. Ali uses 1500 coin and does one upgrade for MAX_HP. Then he starts second level.

Description

Ali passes level and next-level screen comes with the help of FileManager. Ali do upgrades with the help of mouse. After he does upgrade for MAX_HP. Spaceship's doUpgradeWithCoins(MAX_HP) is called and Ali's coins drop to 700 (1700-1000). Then, he chooses to continue with next level, Game's startGame(LEVEL_2, Fighter) is called and GameEngine becomes ready for next level.



Visual Paradigm Standard Edition(Bilkent Univ.)
Ali
GameEngine
FileManager
InputManager
Spaceship
Game
1: passLevel()
1.1: <<void>>
2: Do Upgrade - Increase Max HP
2.1: doUpgradeWithCoins(MAX_HP)
3: Continue with Next Level
3.1: startGame(LEVEL_2, FIGHTER)
3.1.1: <<void>>

Figure 7 – Do Upgrade Sequence Diagram

#4

View Help

Primary Actor: Ali

Scenario

Ali double clicks the game icon in the desktop. Then game opens, first, he chooses to view settings, he changes key for shooting and makes it D key. Then, he returns to menu. Second, he chooses to view high scores. Then, he returns to menu. Third, he chooses to view help. Then he returns to menu.

Description

Ali uses mouse for viewing settings, changing settings, viewing high scores and viewing help in order.

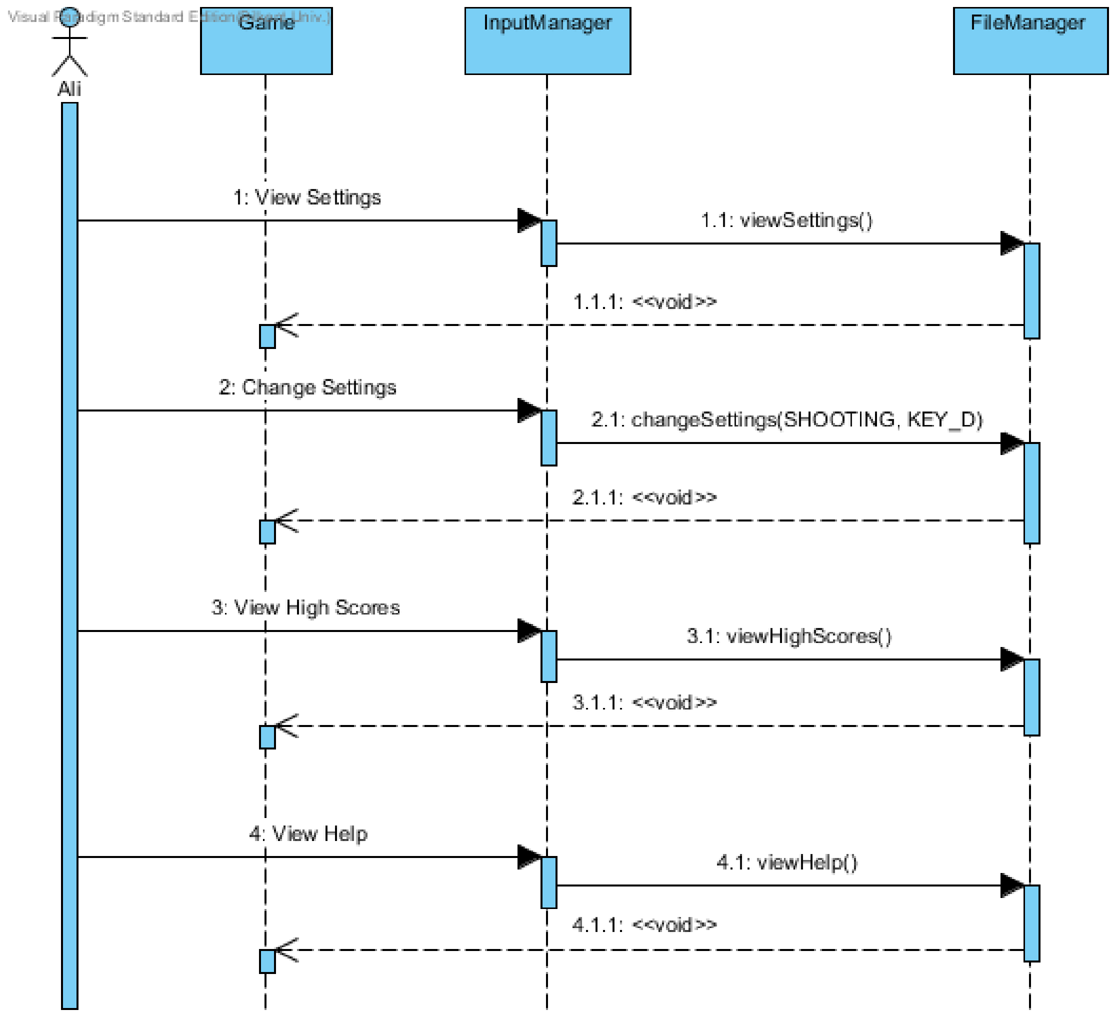


Figure 8 – View Help Sequence Diagram

6.2.2 Activity Diagram

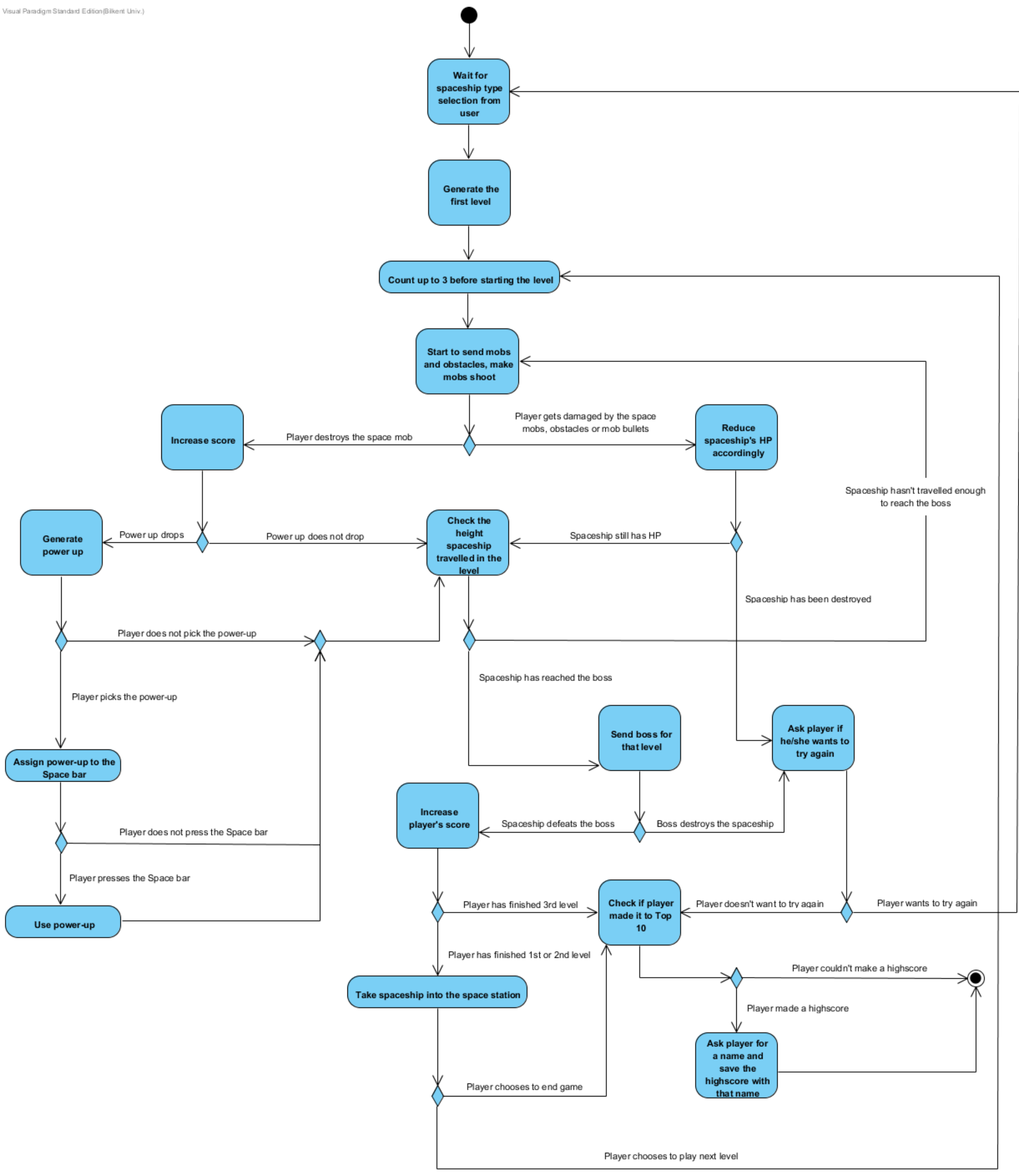

Visual Paradigm Standard Edition(Bilkent Univ.)
Wait for spaceship type selection from user
Generate the first level
Count up to 3 before starting the level
Start to send mobs and obstacles, make mobs shoot
Player destroys the space mob
Player gets damaged by the space mobs, obstacles or mob bullets
Increase score
Reduce spaceship's HP accordingly
Spaceship hasn't travelled enough to reach the boss
Generate power up
Power up drops
Power up does not drop
Check the height spaceship travelled in the level
Spaceship still has HP
Spaceship has been destroyed
Player does not pick the power-up
Spaceship has reached the boss
Player picks the power-up
Send boss for that level
Ask player if he/she wants to try again
Assign power-up to the Space bar
Increase player's score
Spaceship defeats the boss
Boss destroys the spaceship
Player does not press the Space bar
Player presses the Space bar
Use power-up
Player has finished 3rd level
Check if player made it to Top 10
Player doesn't want to try again
Player wants to try again
Player has finished 1st or 2nd level
Player couldn't make a highscore
Take spaceship into the space station
Player made a highscore
Ask player for a name and save the highscore with that name
Player chooses to end game
Player chooses to play next level

Figure 9 – Activity Diagram

Description

Player chooses to play game and selects the spaceship type to play with. System prepares the first level. After three seconds count-down, the game starts. System periodically sends mobs and obstacles. Then, it makes mobs shoot bullets. Spaceship can be damaged by these obstacles, mobs or mobs' bullets. Player's score will increase as time passes. Also, if spaceship destroys mob, score will increase. Mobs may drop power-ups when they are destroyed. Spaceship can pick them and use them with space key. If player successfully reaches to the end of level without dropping spaceship's HP to zero, the unique boss for that level will appear. If spaceship defeats the boss, spaceship will go to the space station. Here, Spaceship's HP will become equal to the max HP. Also, player will be able to do upgrades for spaceship with the coins collected. If the level player passed is 1 or 2, player can continue with next level. If the level is 3, game will finish. Then, system will check whether or not player's score is at top ten. If it is at top ten, player will be asked to give a name for the score gained so the score will be placed to high scores.

6.3 Object and Class Model

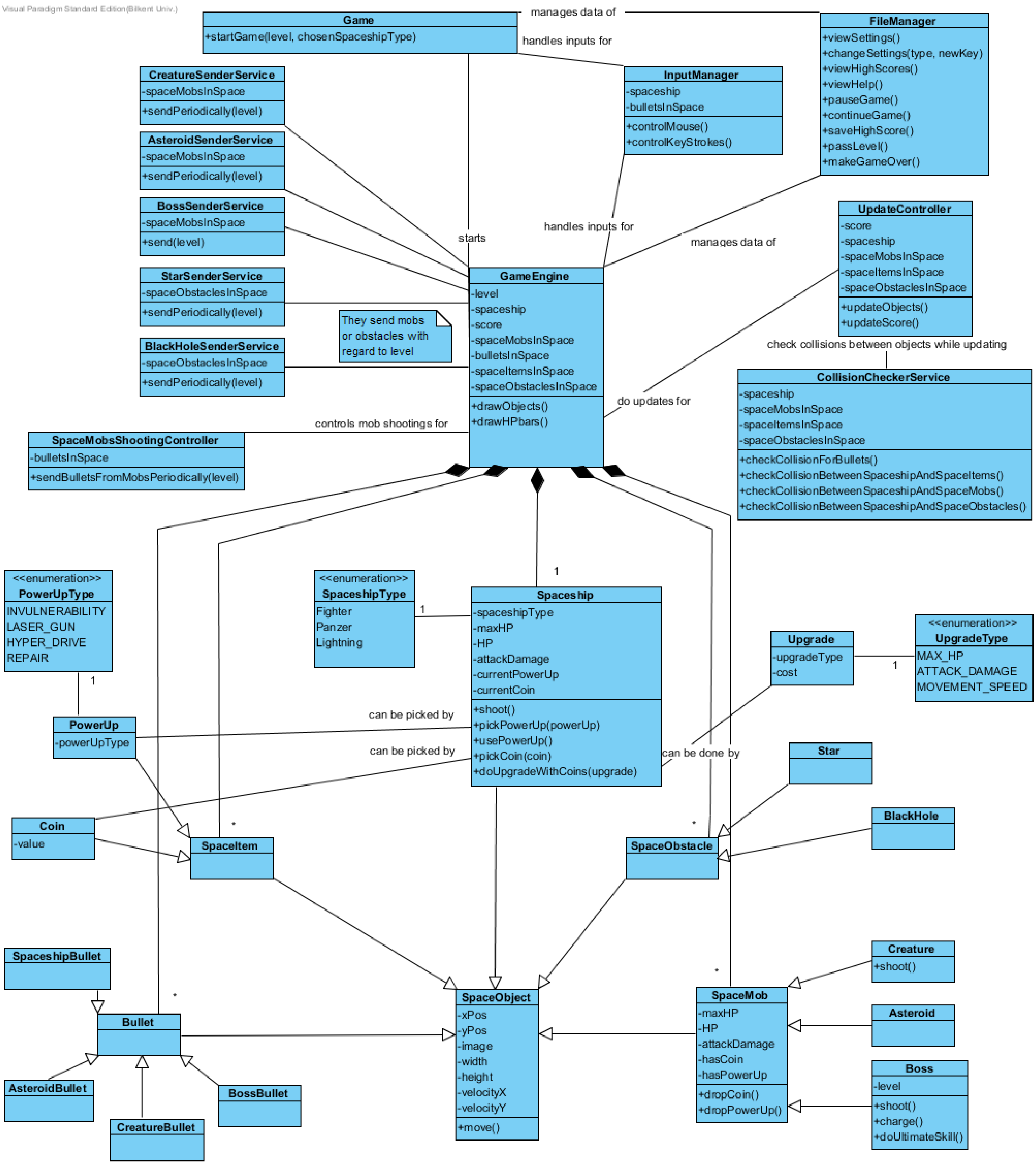


Figure 10 – Object and Class Model

6.4 User interface - navigational paths and screen mock-ups

6.4.1 Navigational Path

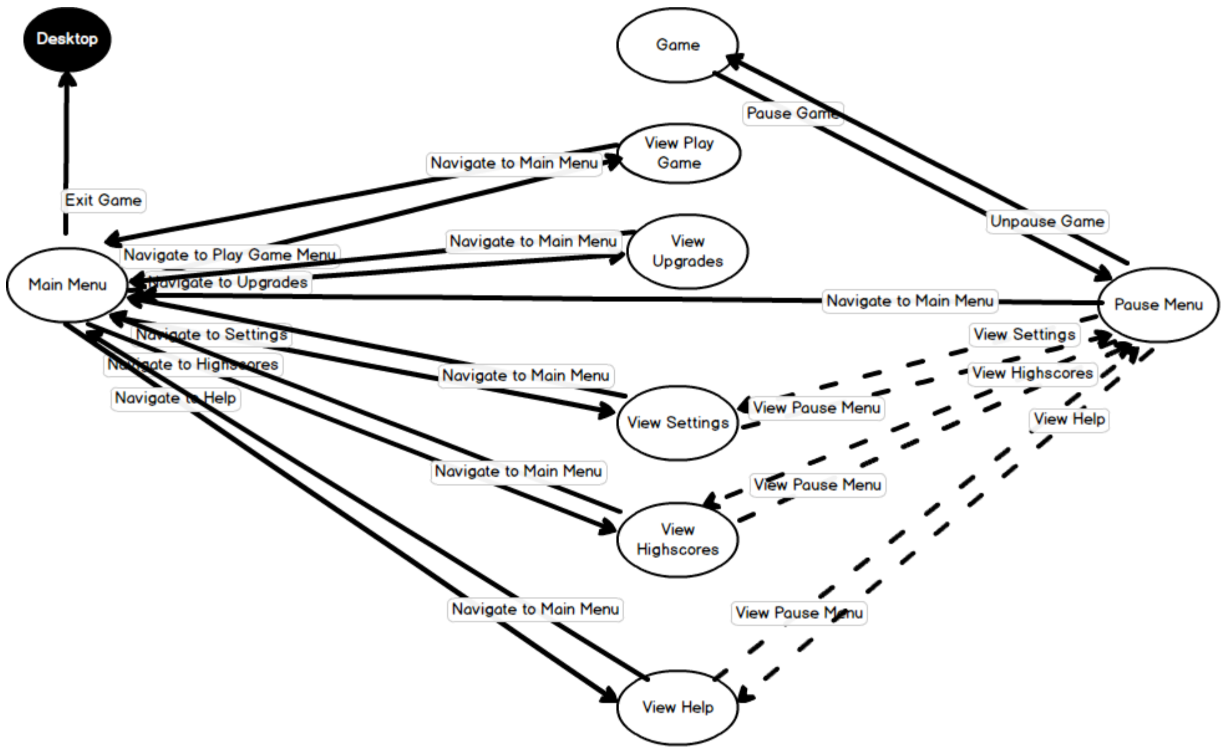


Figure 10 – Navigational Path

Dashed lines in the navigational path denotes that if the player accesses one of the screens which is possible to access from pause menu, back button in the screen will return the player back to pause menu. The same principle applies for the main menu too. In this way, it is not possible for the player to go back to main menu by following the path below:

Game → Pause Menu → View Settings → Navigate to Main Menu

6.4.2 Screen Mock-ups

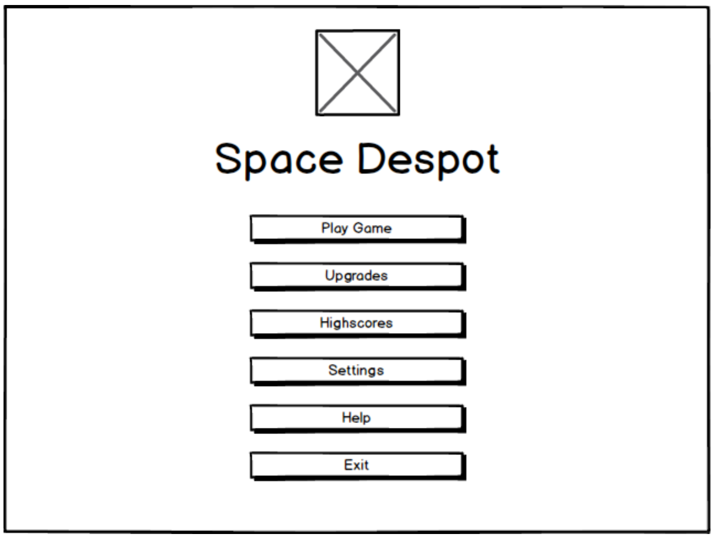


Figure 11 – Main Menu

Figure above is the main menu that the player will see when he opens the game. By clicking the corresponding buttons, the player can access to various screens of the game.

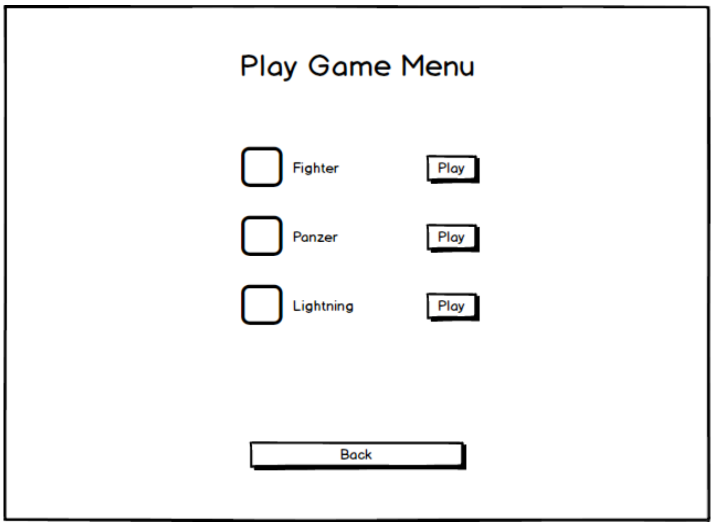


Figure 12 – Play Game Menu

In Play Game Menu, player can select which spaceship he would like to play with and by clicking the corresponding play button, player starts the game.

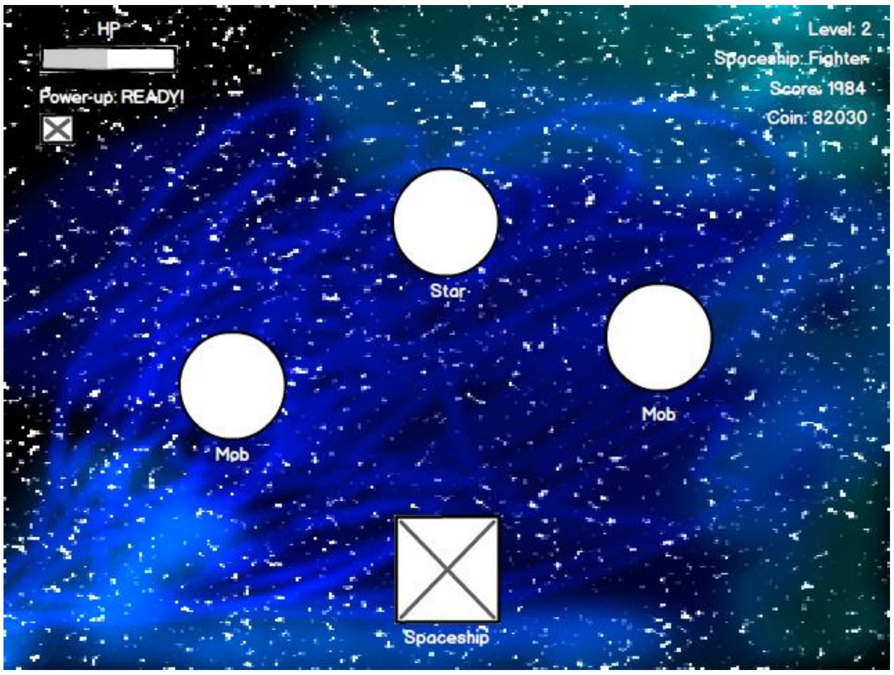
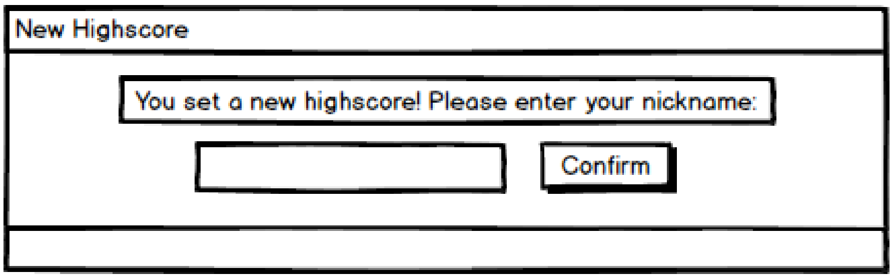


Figure 13 - Game

Above is the game screen where game progresses. Current level, spaceship type, current score and coin amount is shown in the upper-right corner, whereas remaining HP and power-up status and icon is shown in the upper-left corner.


New Highscore

You set a new highscore! Please enter your nickname:

Figure 14 – High Score Achievement Box

If the player achieves a high score, the game will ask for his nickname to save his score to the high scores list.

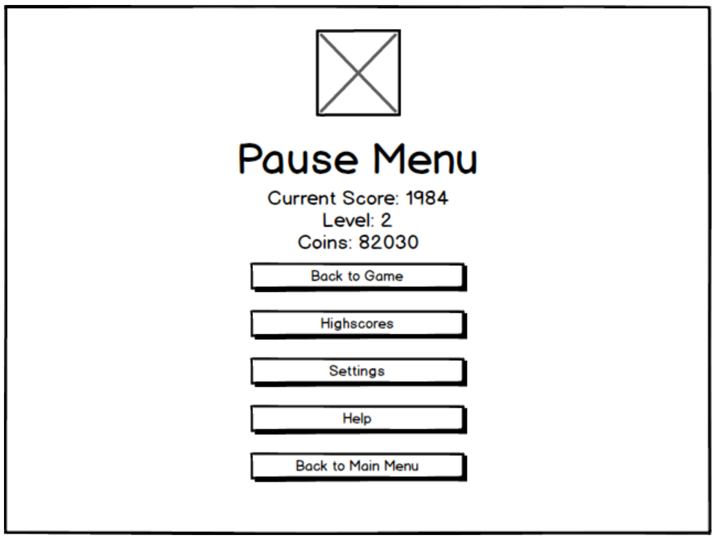


Figure 15 – Pause Menu

Player can access to pause menu by pressing ESC button during the game session.

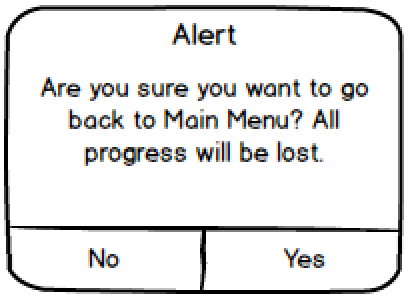
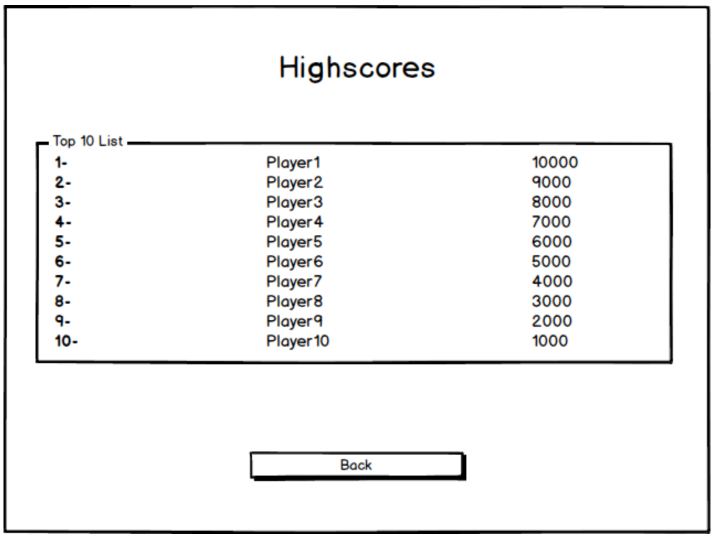

Alert
Are you sure you want to go back to Main Menu? All progress will be lost.
No
Yes

Figure 16 - Alert Box which asks for permission of the player

If the player clicks Back to Main Menu while in Pause Menu, the game will ask for permission of the user whether the player is sure about terminating the game session. If yes is selected, all the progress of the player will be lost, score of the player will not be saved. In this way, it is not possible for the player to accidentally terminate the progress by mis-clicking.



Highscores

Top 10 List

1-	Player1	10000
2-	Player2	9000
3-	Player3	8000
4-	Player4	7000
5-	Player5	6000
6-	Player6	5000
7-	Player7	4000
8-	Player8	3000
9-	Player9	2000
10-	Player10	1000

Figure 17 – High Scores

If the player achieves a high score, his nickname will be seen in high scores screen.

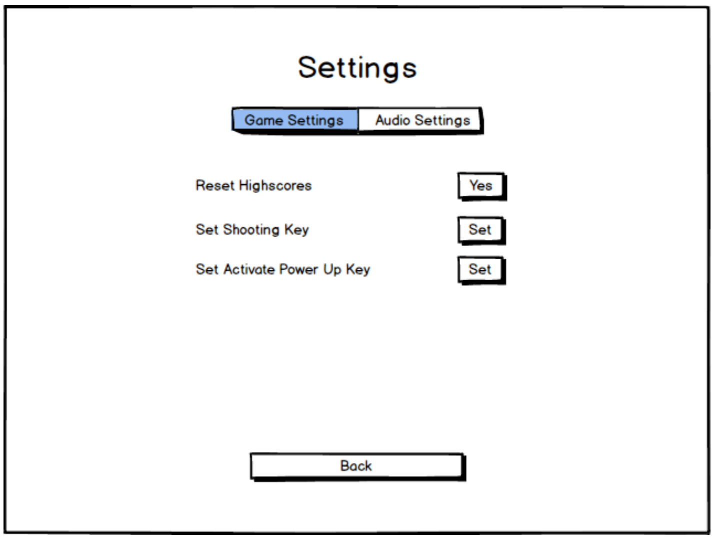


Figure 18 – Settings when Game Settings is selected

Player can reset high scores, set shooting key and activate power up key in the settings screen.

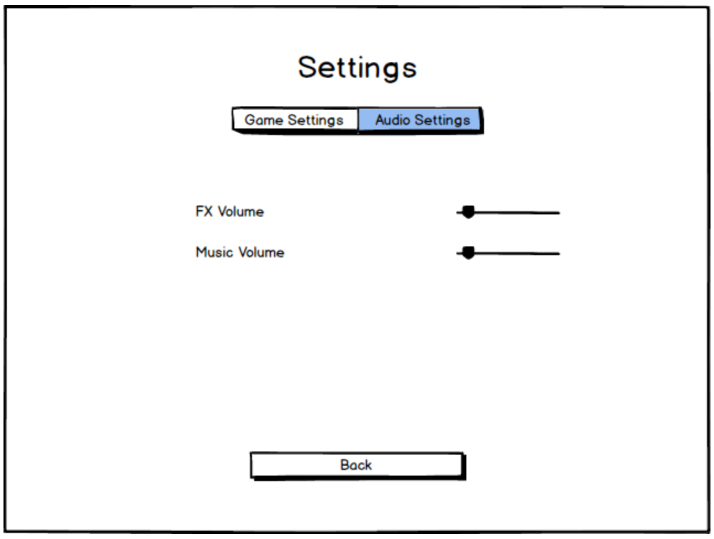


Figure 19 – Settings when Audio Settings is selected

Player can set FX Volume and Music Volume by navigating to Audio Settings screen in Settings screen.

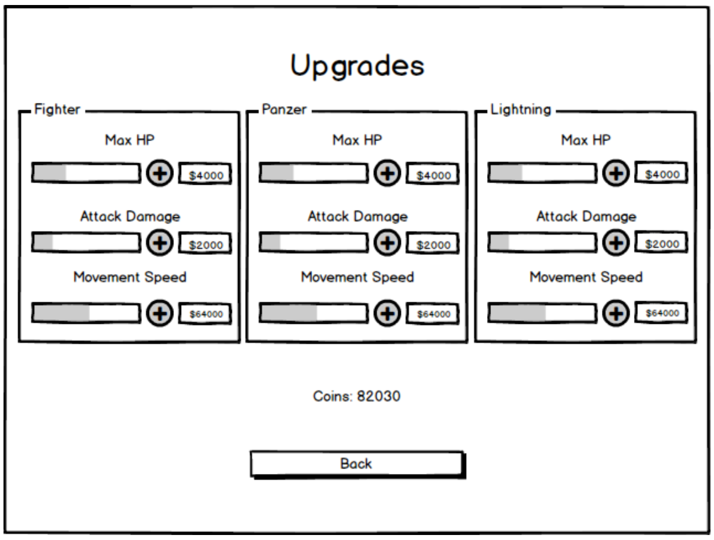
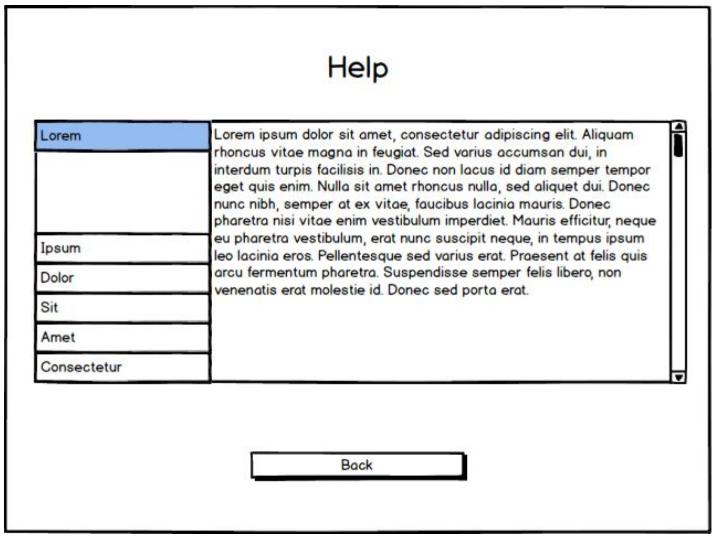


Figure 20 - Upgrades

Player can buy upgrades to the spaceships in upgrades screen. The progress of the each upgrade and required money for the next upgrade phase is shown below the title of each upgrade.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam rhoncus vitae magna in feugiat. Sed varius accumsan dui, in interdum turpis facilisis in. Donec non lacus id diam semper tempor eget quis enim. Nulla sit amet rhoncus nulla, sed aliquet dui. Donec nunc nibh, semper at ex vitae, faucibus lacinia mauris. Donec pharetra nisi vitae enim vestibulum imperdiet. Mauris efficitur, neque eu pharetra vestibulum, erat nunc suscipit neque, in tempus ipsum leo lacinia eros. Pellentesque sed varius erat. Praesent at felis quis arcu fermentum pharetra. Suspendisse semper felis libero, non venenatis erat molestie id. Donec sed porta erat.

Figure 21 - Help

By navigating to Help screen, player can learn about the game features, controls and get knowledge and tips about upgrades, how to play better etc.

7 Glossary

HP: Health Points

FX: Effects

8 References

[1] Space Impact. https://en.wikipedia.org/wiki/Space_Impact [Accessed: Oct 15, 2016].