Bilkent University

Department of Computer Engineering

CS 319
Object Oriented Software Engineering

Analysis Report

Space Invaders

Group10 Members:

Ayşe Öykü Özer

Turgay Arda Usman

Yonca Yunatcı

Table of Contents

1. Introduction

The project we are planning to implement is the game called "Space Invaders". Different from the original game, our version will have additional properties. Original game can be found at:

http://www.oyunlar1.com/games.php?flash=603

We chose this game since it is a simple beloved game. It does not require complex algorithms instead, it requires high number of classes to implement and different types of relationships to make. We plan to focus on object oriented design rather than algorithms. Therefore, we believe that it is a proper project for the course CS319.

2. Overview

The game has the purpose of destroying the aliens while trying to keep the spaceship alive. When the user starts the game from the main menu, there are two main objects: a block of aliens and the spaceship. In the game user can control the spaceship and tries to eliminate the whole block of aliens by the laser gun. If it hits an alien, the closer one gets destroyed. On the other hand, alien block also bombs the spaceship. Player begins the game with just one life; therefore, if a bomb hits the spaceship game will be over immediately. For simplicity, spaceship is represented as a bar and aliens as just their heads.

2.1. Controls

The only thing the user can control is spaceship and the time when it sends beams to the aliens. Spaceship moves horizontally and controlled by arrow keys on the keyboard. Attack is controlled by the space bar.

2.2. Levels

There are two options of the game according to its difficulty level: easy and hard. For the each level, the speed of the aliens increases. Moreover, for each of the 2 levels, aliens become harder to kill. After the level 2, aliens will have more than one life; thus, it takes more than one shot to kill them.

2.3. Coins and power-ups

Player earns coins and points during the game by destroying the aliens. At the end of the each successful game player can improve the spaceship by spending the coins earned. Improvements include "attack damage", "speed" and "beam speed".

Attack damage

As mentioned above, for the further levels, aliens become harder to kill. This power-up enables the player to destroy the aliens easily compared to the previous levels.

Speed

If this power-up is purchased, the speed of the spaceship will increase. It will horizontally move faster which is a necessary skill to avoid the bombs.

Beam speed

The beam that goes from the spaceship to the aliens should reach at the right time in order to destroy the aliens successfully. Since only one beam

can be sent at a time, this power-up not only increases the speed of the

beam but also offers the player to attack more often.

2.4. Other Features

The game proposes 4 other options such as "High Scores", "Help" and

"Settings" and "Credits". High scores will list the top 10 scores with the

usernames. The total points earned from killing the aliens and passing the

levels will be shown in this section. Help button will explain the controls and the

rules of the game in order to be sure that user plays the game properly. From

the settings option, the user can either disable or enable the sound. Finally

credits section displays the list of the contributors.

3. Functional Requirements

3.1. Play Game

User begins the game with one life and tries to kill the aliens by shooting

upwards. Aliens slide horizontally, when they reach one side of the application

window, they come one step closer to the spaceship that user controls. The

spaceship can slide horizontally at the bottom of the application window. There

are no boundaries for the spaceship as opposed to the aliens. If the spaceship

reaches left-most side and user continues to push the left button, the spaceship

appears in the right-most side of the window and vice versa. User gains coins

and points for each alien that the he or she kills. Aliens throw bombs to the

ground to kill the user. If one of the bomb touches the spaceship, game will be

over.

3.1.1. Easy Mode

In the easy mode, the direction of the bombs coming from the aliens is vertical. No directional change will occur once the bomb is released.

3.1.2. Hard Mode

In the hard mode, bombs released from aliens follow the spaceship. While the bomb is falling, the game checks two times where the spaceship is and determines a new direction for the bomb if the spaceship is not at the same vertical line with the bomb.

3.2. Next Level

If all aliens are killed, current level is succeeded. Before playing the next level, user have options to upgrade the spaceship's qualifications. Next level proposes a higher speed of aliens and it becomes harder to kill the aliens. Moreover, the number of bombs released from aliens will increase. In order to be successful in the further levels, user can be upgraded in 3 fields: Attack damage, speed of the bullets and the speed of the spaceship.

3.3. End of the game

If one of the bomb or one of the alien touches the spaceship game will be over. At the end of the game, a window will appear which asks for the name to submit the score if the score is higher than the existing top 10 scores. User can enter his/her score by typing username. By default username is "Player". Otherwise, game will just display a message that informs the player that the game is over.

3.4. Pause Game

User can pause the game for that particular moment. After pushing the pause button, 2 options can be chosen. User can either continue from where he/she left or exit the current game without saving and return to the main menu.

3.5. Exit Game

If the user pushes the exit button the application window will be closed. If the user wants to exit the game while playing the score will be lost.

3.6. High Scores

Top 10 scores will be demonstrated with usernames.

3.7. Help

The instructions to play the game will be explained. The instructions contains two parts: First, the control buttons on the keyboard will be specified. Secondly, the rules and the process will be explained in details.

3.8. Credits

The list of the contributors will be shown.

3.9. Settings

Sound can be enabled or disabled.

4.Non-functional Requirements

4.1. Convenience of Controls

The game should be easily controlled. The selection from the main menu should be done by clicking with mouse or touchpad. For the playing part of the game keyboard is the best option. For the convenience, the common controllers used in this type of games will also be used in this project too.

4.2. Performance

The input from keyboard should be processed less than half a minute in order not to lose any time. Response time is a crucial component to provide a proper gaming experience. For example, user tries to move the spaceship but if this input were processed lately, a bomb might touch the spaceship which would cause the game to end.

4.3. Graphics

The animation of the game should be smooth in order to provide a better visualization. Furthermore, graphics should be as simple as possible to avoid the chaos that might occur when things get confused with each other.

4.4. Extendibility

Development is an important issue in the game "Space Invaders". In order to add new properties and functions, implementation should be extendable. For example, the implementation should be compatible to support multiplayer game.
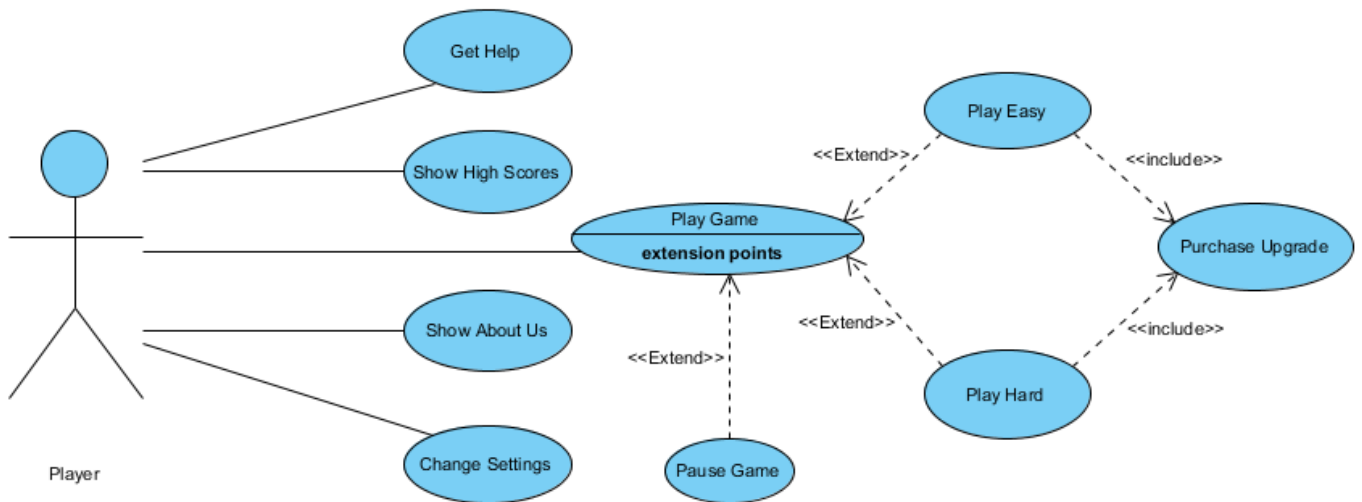
5. System Models

5.1. Use Case Model

Figure 1: Illustrates the use case model of the game

5.1.1. Play

Use Case Name: Play

Primary Actor: Player

Stakeholders and Interests:

- Player: pushes the play button to begin the game. The aim is to gain coins and points by trying to stay alive.

- Play: should open the window in which the level is chosen.

- Easy: If the easy button is pushed, an easy game should be opened.

- Hard: If the hard button is pushed, a hard game should be opened.

- High Scores: should keep the track of the scores and record if it is one of the highest 10 scores.

Pre – condition: Player should currently view main menu

Post – condition: Score should be kept if it is more than current 10 scores.

Entry condition: Player should choose "Play" button from the main menu.

Exit condition: One of the following conditions should be occurred:

- Player loses the game.

- Player should choose "Exit" button in the game window. Then choose one of the options which are: returning the main menu and quitting the game.

Main Flow:

a) Player initiates the game

b) According to the choose of level, easy or hard game will be loaded by the system.

c) Player tries to kill the aliens and avoid bombs in order to gain points and qualify to the next level.

d) At a certain level, player gets hit and the game will be over.

Alternative Flows:

1. Submit the high score

    a) At the end of the game, if the total score is greater than the current top 10 score, a window which asks for the name will appear in order to submit the high score.

    b) The player enters a username.

    c) The system will update the high score list with the new score and the given username.

2. Player presses the "Pause" button

    a. Pause menu will appear.

  b. Three options can be chosen: continue on playing, returning the main menu or disable/enable sound.

  c. If the player chooses to return the main menu, current game will be lost and score will not be kept.

3. A power-up is chosen

  a. After the level is succeeded, a power-up can be purchased.

  b. If Attack damage is purchased with coins, player gains more ability to destroy the aliens.

  c. If speed is purchased with coins, spaceship will move faster.

  d. If beam speed is purchased with coins, the beams to kill the aliens will have a higher speed.

  e. After the purchase, the system will upgrade the amount of coins remain.

  f. Player can continue to play the next level by clicking "continue" button.

5.1.2. Help

Use Case Name: Help

Primary Actor: Player

Stakeholders and Interests:

- Player: pushes the help button to learn more about how to play the game.

- Help: shows instructions and controllers to play the game

Pre – condition: Player should currently view main menu

Post – condition: -

Entry condition: Player should choose "Help" button from the main menu.

Exit condition: Player should choose "Back" button in the Help window.

5.1.3. High Scores

Use Case Name: High Scores

Primary Actor: Player

Stakeholders and Interests:

- Player: opens the high score page in order to see his/her ranking or view the top 10 scores with the usernames.

- High Scores: views the list of top 10 scores.

Pre – condition:

- Player should currently view main menu

- The highest 10 scores should be recorded beforehand.

Post – condition: -

Entry condition: Player should choose "High Scores" button from the main menu.

Exit condition: Player should choose "Back" button in the High Scores window.

5.1.4. Credits

Use Case Name: Credits

Primary Actor: Player

Stakeholders and Interests:

- Player: wants to see the list of the contributors.

- Credits: shows the list of the contributors.

Pre – condition: Player should currently view main menu

Post – condition: -

Entry condition: Player should choose "Credits" button from the main menu.

Exit condition: Player should choose "Back" button in the Credits window.

5.1.5.Exit

Use Case Name: Exit

Primary Actor: Player

Stakeholders and Interests:

- Player: wants to quit the game.

- Exit: If currently the playing window is open, presents the opportunity to go back to the main menu. If the player is already in the main menu, program should terminate.

Pre – condition: -

Post – condition: -

Entry condition: Player should press the "Exit" button.

Exit condition: -

5.2 Dynamic Models

5.2.1 State Chart
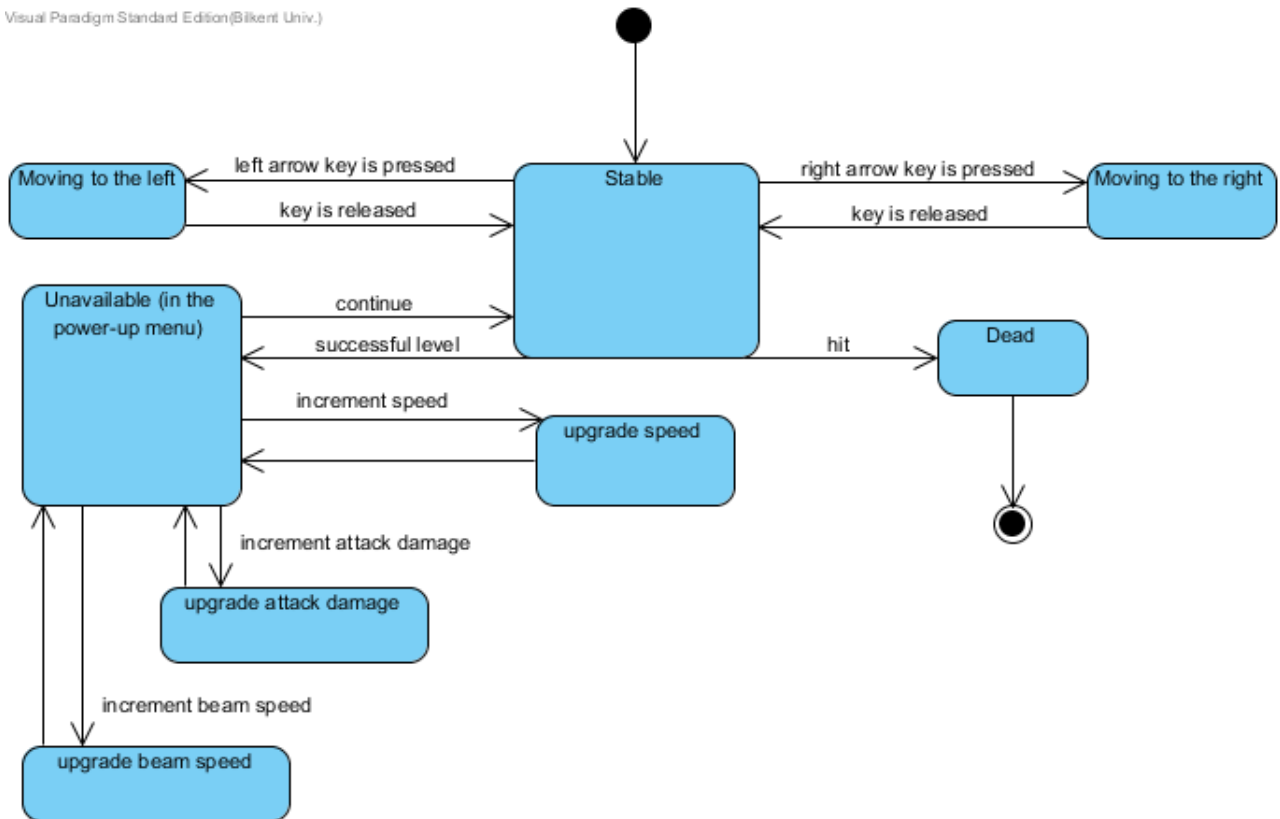
States of the player

Figure 2: Illustrates the state chart diagram for the player

This state diagram explains the behavior of the player. If the player uses user keys, the player will move horizontally along the screen. The player will also be able to pass through the screen borders that are when player reaches the end of the left side of the screen, player will automatically passes to the right end of the screen. If there is no input, it is initially at the "stable" state. After a successful level, player is in inactive mode namely "unable". Player can either purchase power-up or continue to the next level. If the player gets hit, game will be over.
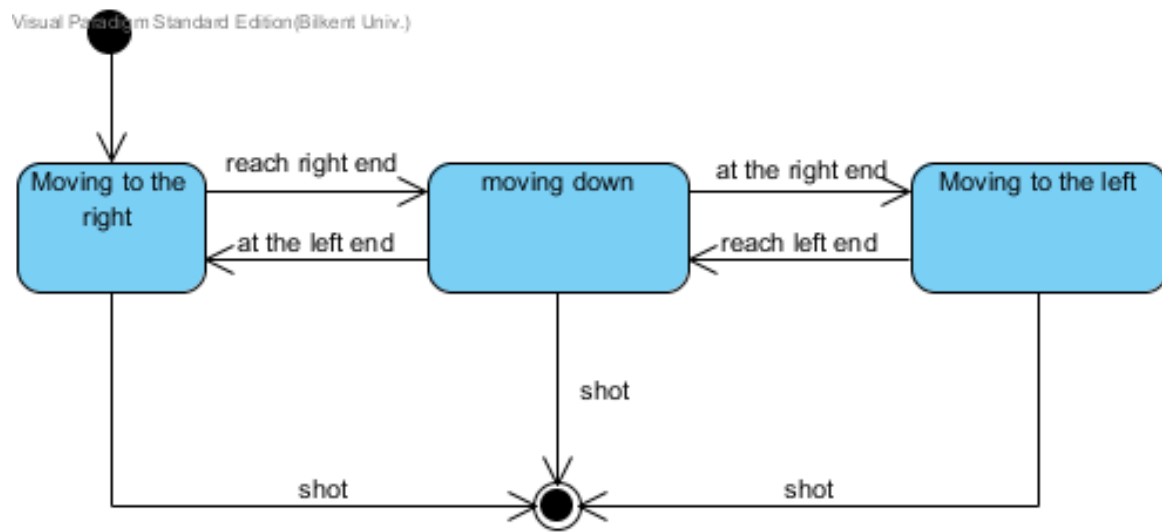
States of the alien

Figure 3: Illustrates the state chart diagram for the alien

Figure 3 displays the states of an alien object. Alien initially move to the right but when it reaches at the right end of the screen, it will vertically move one step closer to the spaceship. Then, it continues to move to the left. When it reaches at the left end of the screen, the same rule will be applied. Alien can be shot during any of these states. This case is the end of the life of an alien.
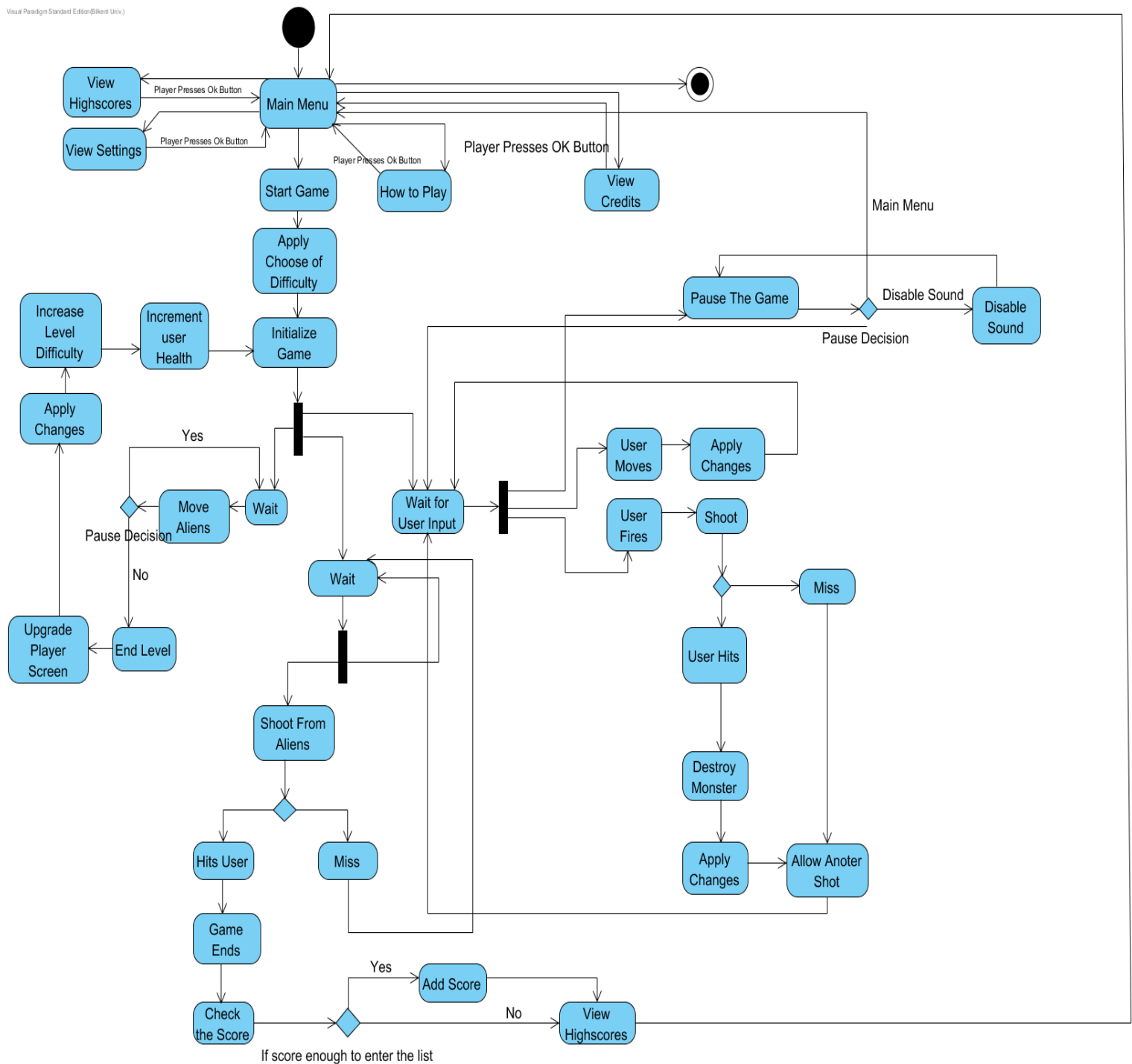
5.2.2 Activity Diagram:

Figure 4: Illustrates the activity diagram for the game

The activity diagram above shows the main flow of the game. Game is initialized from the main menu and system presents some options such as viewing high scores, settings, credits or how to play. The system can go back to the main menu after selecting one of the listed options. The system can terminate with the quit option or start a new game. If a new game is started system will ask the difficulty level and apply the changes. The game is initialized from the first level and starts to wait input from the player. If player uses the arrow keys, the system allows the user to move horizontally along the screen. If the player chooses the space key the system fires a laser beam. If the beam hits an alien, the system destroys that alien but the laser beam can also miss the alien. In that case, the system does not destroy the alien. In both cases system allows player to shoot another shot if the game is not finished. Meanwhile, the system waits for random seconds according to the difficulty of that level and moves the aliens and checks if the aliens have reached the end of the screen which causes the player to lose. Moreover, the system makes aliens to throw bombs randomly. If a bomb hits the player, the player, again, loses the game. Another possibility occurs when the esc key is pressed. If that happens the pause menu is displayed, the system will terminate from this screen by pressing the esc button again.

At the end of the game the system checks the score. If the score is greater than any of the scores which are on the list, the system rearranges the scores and then views it to the player. If the score is not able to get into the high score list, the system just displays the current high scores.

5.2.3 Sequence Diagram:

Scenario 1 - Help:

- The player chooses "Help" option from the main menu, to learn how to play the game.

- Help is shown on the screen.

- If the player wants to return to the main menu, he/she chooses the "Back" option of the game manual and exits from the manual.
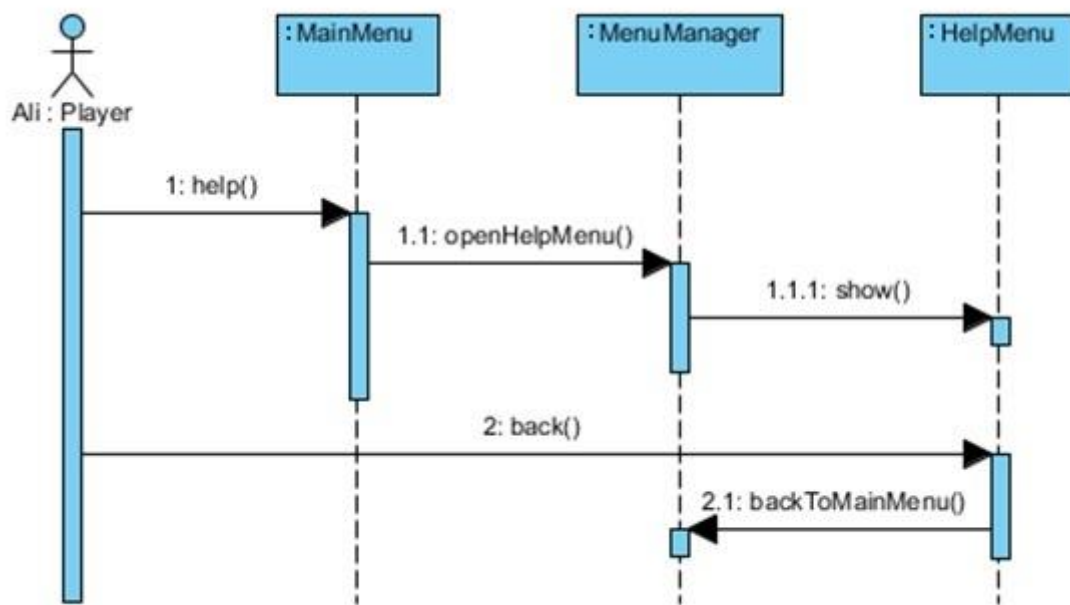

Figure 5: Shows the sequence diagram for scenario 1

Scenario 2 - High Scores:

- In order to see the top ten scores, the player chooses "View High Scores" option.

- The system shows the highest ten scores.

- If the player wants to return to the main menu, he/she chooses the "Back" option of the top ten score screen and exits from the high scores page.

The sequence diagram for this scenario is almost the same with the Figure 5. Only difference is in the method 1.1: openHighscores() and in the class name:HighscoreMenu

Scenario 3 - Credits:

- Player wants to show credits. View credits option is chosen from the main menu. The system displays the credits.

-Back option is used for going back to main menu from the view credits option.

The sequence diagram for this scenario is almost the same with the Figure 5. Only difference is in the method 1.1: openCredits() and in the class name: CreditsMenu

Scenario 4 - Change Settings:

- Settings can be changed from the main menu or from the pause menu.

- Settings include sound level and music of the game. Sound level can be changed, turned off and music can be changed from the settings.

-Settings are updated by the changed made by player.

- Player uses the back menu to return to main menu or pause menu from where change settings menu is reached.
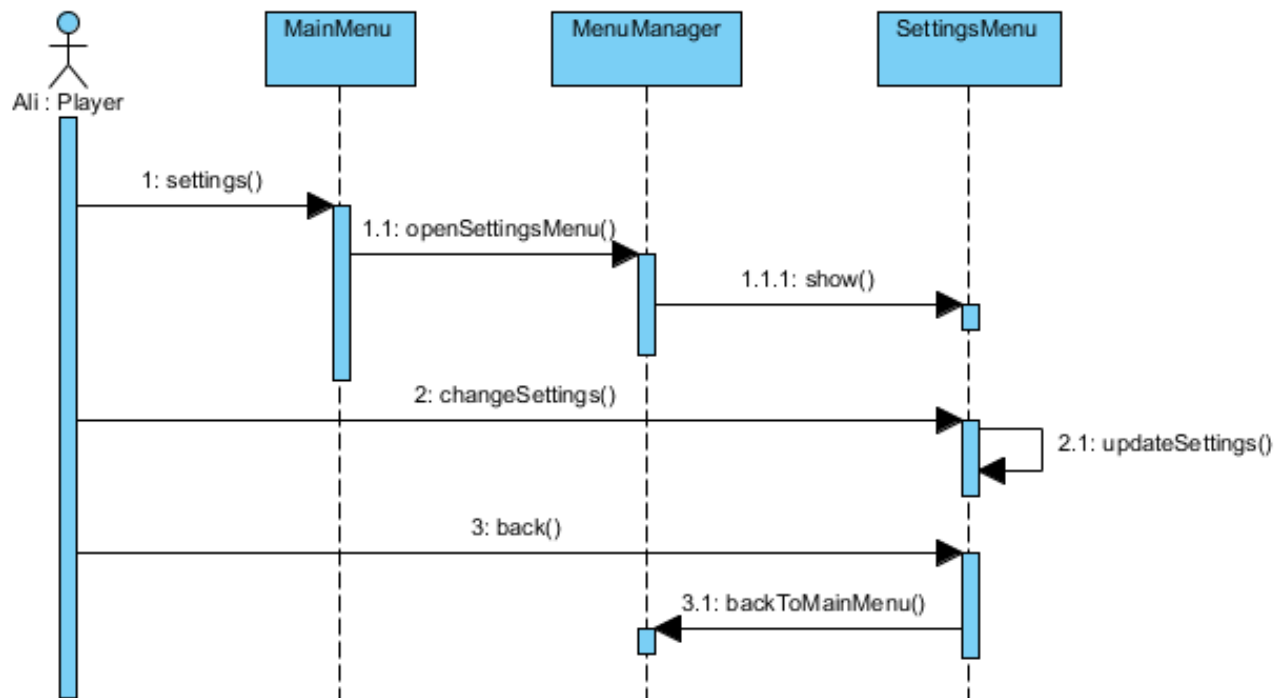
Figure 6: Shows the sequence diagram for scenario 4

Scenario 5 - Power-Ups:

-At the end of the each successfully completed level, the system gives two options.

Player can continue game without spending any coins or player can upgrade the

spaceship by spending earned coins.

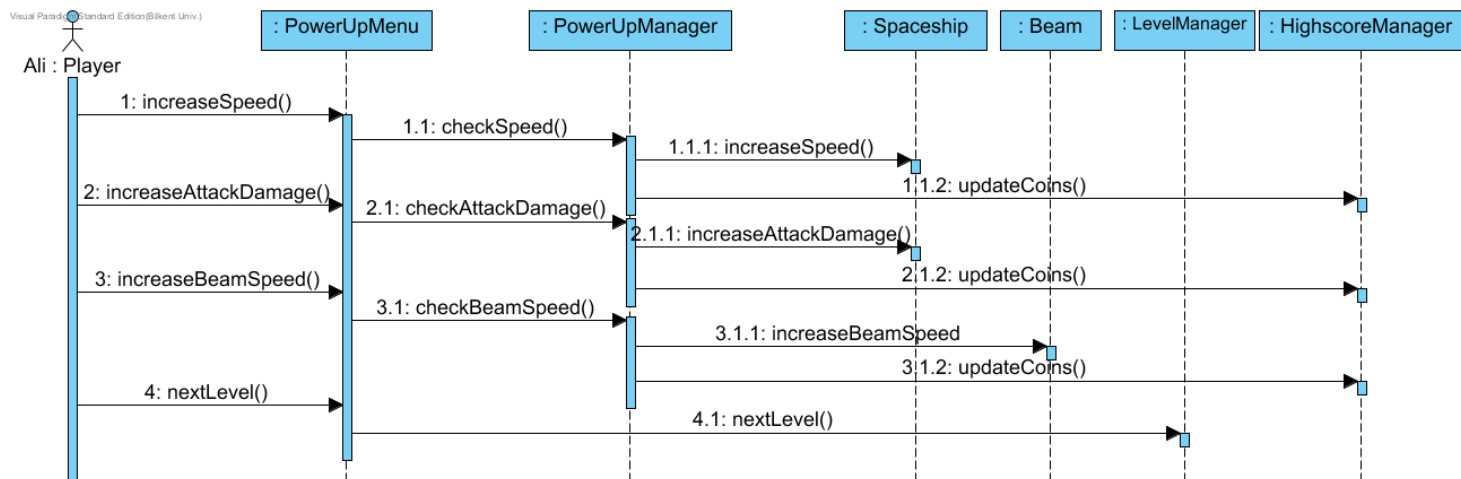-If coins are spent, system activates the remaining coin amount.

Figure 7: Shows the sequence diagram for scenario 5

Scenario 6 - Choose Level:

- The user selects the play game option of the main menu.

-The system asks user the game difficulty level of the game easy or hard.

- The game is loaded according to the game settings and the difficulty level.
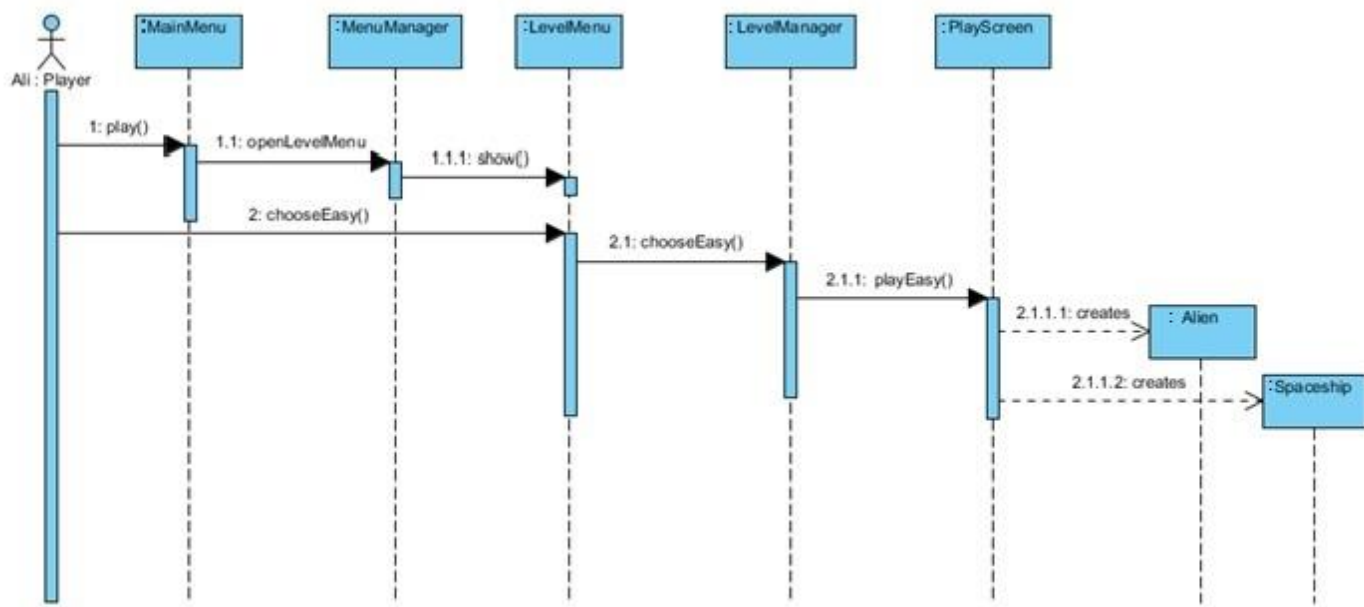
Figure 8: Shows the sequence diagram for scenario 6

Figure shows the interaction between classes when the easy mode is chosen. If the

hard mode was chosen, the only difference will be at the methods number 2, 2.1 and

2.1.1. These method headers will be change into "hard" instead of "easy".

Scenario 7 - Shooting an Alien:

-During the play, player tries to kill the aliens by shooting.

-When the player succeeds aliens status should be updated.

-Player earns points.

-When all the aliens are killed, power up menu should be displayed.

Figure 9: Shows the sequence diagram for scenario 7

Scenario 8 - Game Over:

-Player gets hit by the aliens.

-Game will be over.

-High score will be submitted.

-High score table will be shown.

Figure 10: Shows the sequence diagram for scenario 8

5.3 Object and Class Model

LevelManager
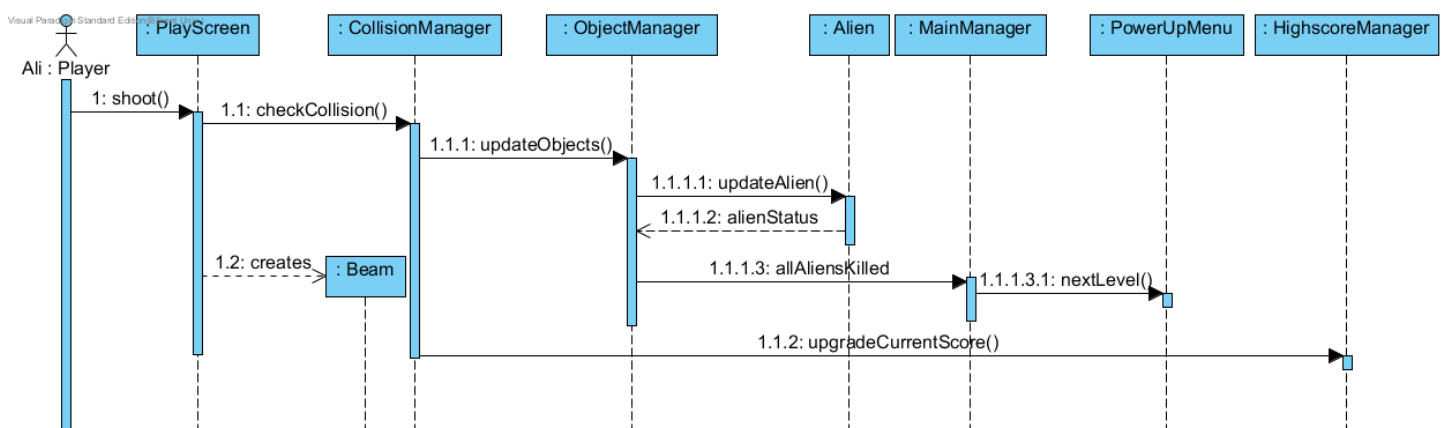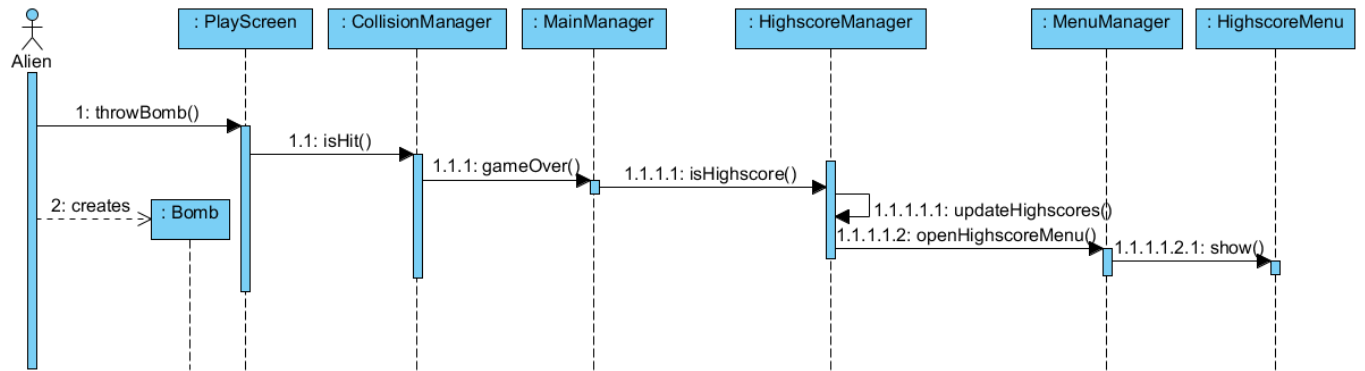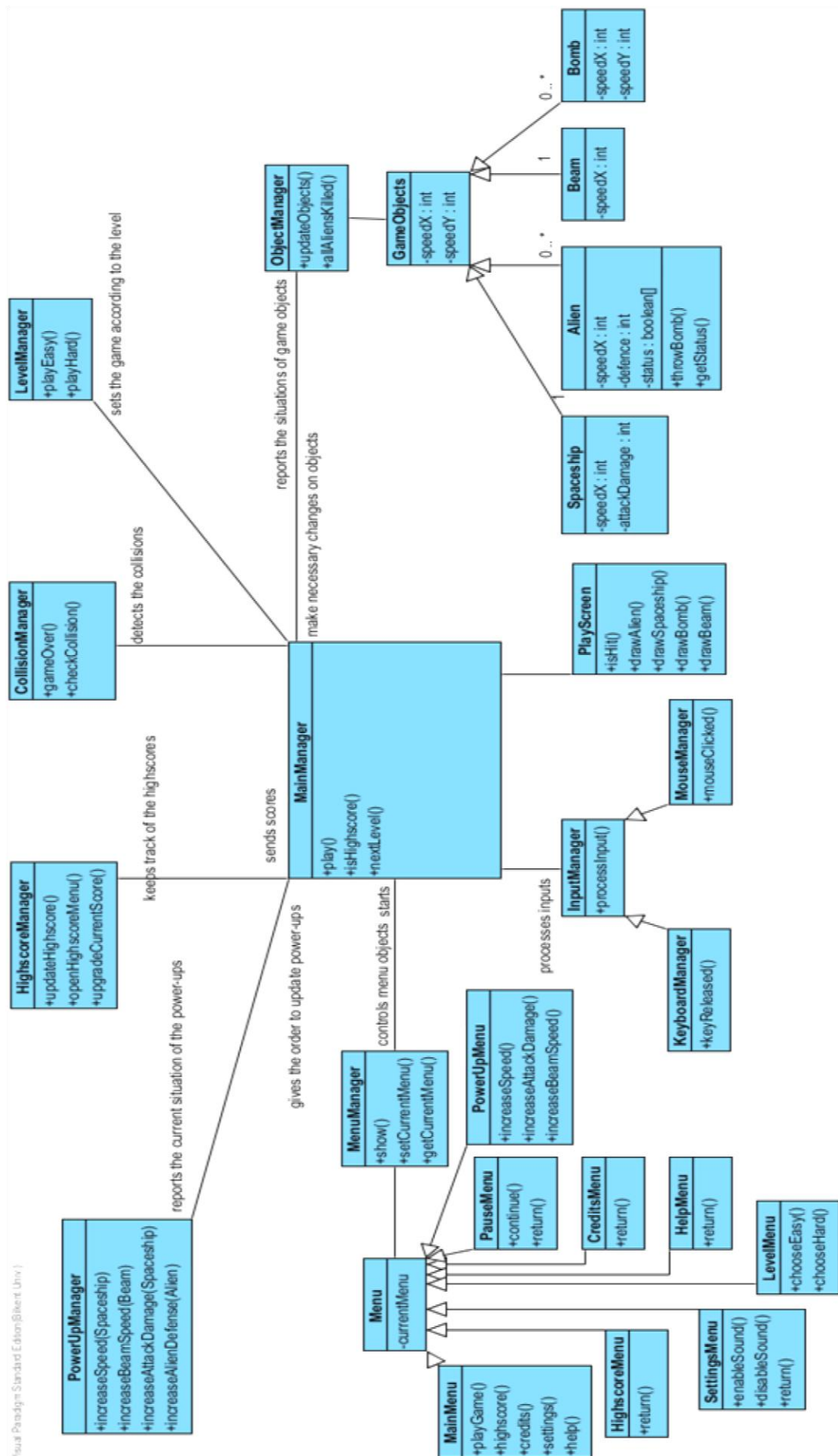+playEasy()
+playHard()

sets the game according to the level

ObjectManager
+updateObjects()
+allAliensKilled()

reports the situations of game objects

GameObjects
-speedX : int
-speedY : int

Beam
-speedX : int

1

Bomb
-speedX : int
-speedY : int

0..*

Alien
-speedX : int
-defence : int
-status : boolean[]
+throwBomb()
+getStatus()

0..*

Spaceship
-speedX : int
-attackDamage : int

make necessary changes on objects

CollisionManager
+gameOver()
+checkCollision()

detects the collisions

MainManager
+play()
+isHighscore()
+nextLevel()

sends scores

keeps track of the highscores

HighscoreManager
+updateHighscore()
+openHighscoreMenu()
+upgradeCurrentScore()

PlayScreen
+isHit()
+drawAlien()
+drawSpaceship()
+drawBomb()
+drawBeam()

InputManager
+processInput()

processes inputs

MouseManager
+mouseClicked()

KeyboardManager
+keyReleased()

MenuManager
+show()
+setCurrentMenu()
+getCurrentMenu()

controls menu objects starts

gives the order to update power-ups

reports the current situation of the power-ups

PowerUpManager
+increaseSpeed(Spaceship)
+increaseBeamSpeed(Beam)
+increaseAttackDamage(Spaceship)
+increaseAlienDefense(Alien)

PowerUpMenu
+increaseSpeed()
+increaseAttackDamage()
+increaseBeamSpeed()

PauseMenu
+continue()
+return()

CreditsMenu
+return()

HelpMenu
+return()

LevelMenu
+chooseEasy()
+chooseHard()

Menu
-currentMenu

MainMenu
+playGame()
+highscore()
+credits()
+settings()
+help()

HighscoreMenu
+return()

SettingsMenu
+enableSound()
+disableSound()
+return()

MainManager is the main control of the whole system. All other controllers report the situations and changes in the area that they are responsible for. MainManager starts the boundary object "Menu". MainMenu, PauseMenu and PowerUpMenu are derived by this parent Menu class. These menu objects are boundary objects that enable the user to communicate with the system.

All manager-named classes are control objects for a particular field. InputManager is the control object for both keyboard and mouse inputs. It processes these inputs and reports to the MainManager. ObjectManager is responsible for all game objects such as spaceship, alien, beam and bomb. It can update the current situation of the objects according to the instructions given by MainManager. PowerUpManager is responsible for receiving the order and if it is appropriate, gives the order to update the power-up. LevelManager sets the difficulty level. CollisionManager detects the collisions and reports to the MainManager. HighscoreManager checks the score given by MainManager whether the score is able to get into the high score table.
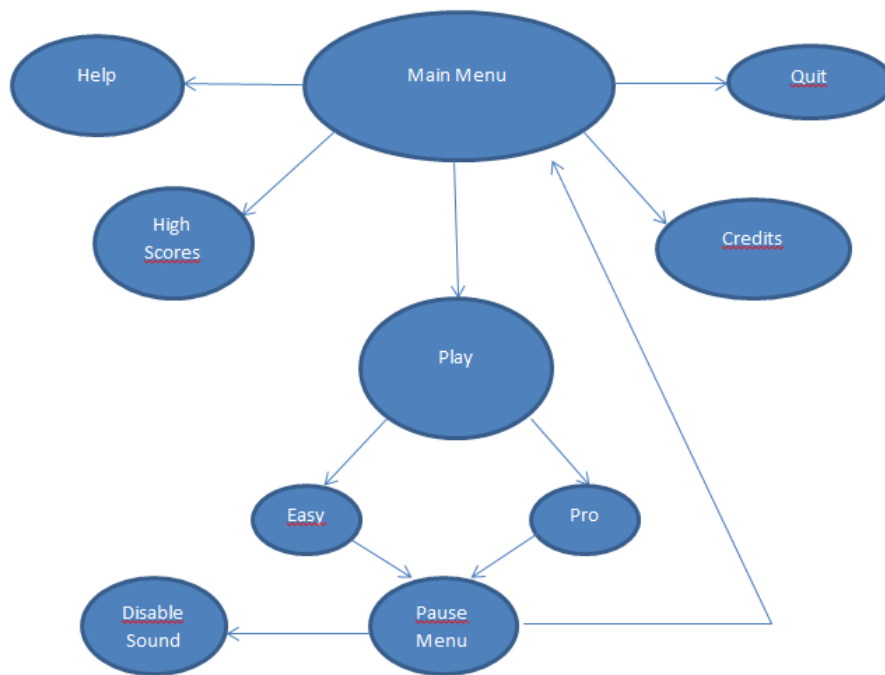
5.4 User Interface

5.4.1 Navigational Path

Figure 12: Illustrates the Navigational Path

5.4.2 Screen Mock-ups:

Main Menu:

When player opens the game, player will see main menu screen. Main menu screen has five options. They are play, high scores, credits, help, and quit.

-Quit: When player selects quit, game stops running and game window is closed**.**

-Play Game: If Player selects "Play", game gives to options to player about the difficulty level of the game: Easy and Hard Figure 5.2.1.1 is the Play option screen.
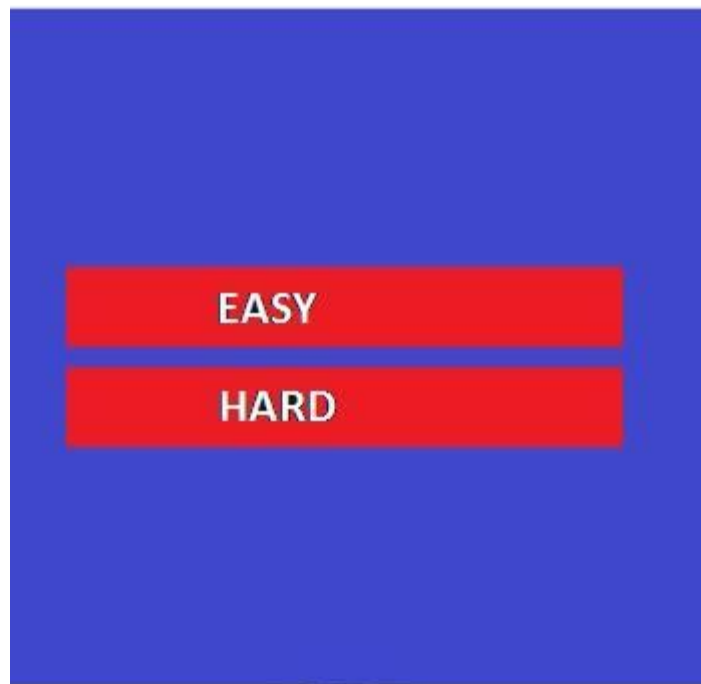

Figure 14: Screen view of play options

    -Easy: If Player selects Easy, game starts with current game settings. Figure 15 is a screen shot from easy option.

Figure 15: Screen view of play with easy option

-Hard: If Player selects Hard, game starts with current game settings. Figure 16 is a screen shot from hard option.

Figure 16: Screen view of play with hard option

-View High Scores: When high scores option is chosen, system will show the list of top ten scores with player names. If player can make a score which is higher than the 10th score, player's score is displayed at this list and he will enter his name to high scores list.

Figure 17: Screen shot of View High Scores

-Help: If player chooses help, a text is shown on screen containing instructions about game. Player can return to pervious menu by choosing "Main Menu".

Figure 18: Screen shot of Help

-Credits: When Player selects Credits, names of game developers are shown on screen. Player can return to previous menu by choosing "Main Menu".

Figure 19: Screen shot of Credits

-End of Successful Level: At the end of the each successful level, before starting the new level, game shows the total coins the player have and three power ups that player can buy as long as player has enough coins.

Figure 20: Screen between two consecutive levels.

Pause Menu

When Player pauses game by pressing "ESC" on keyboard, pause menu will be shown on screen. Pause Menu has "Resume", "Main Menu" and "Disable/Enable Sound" options..

-Resume: This option continues the game from where player pauses the game.

-Main Menu: This option ends the game and take player to main menu.

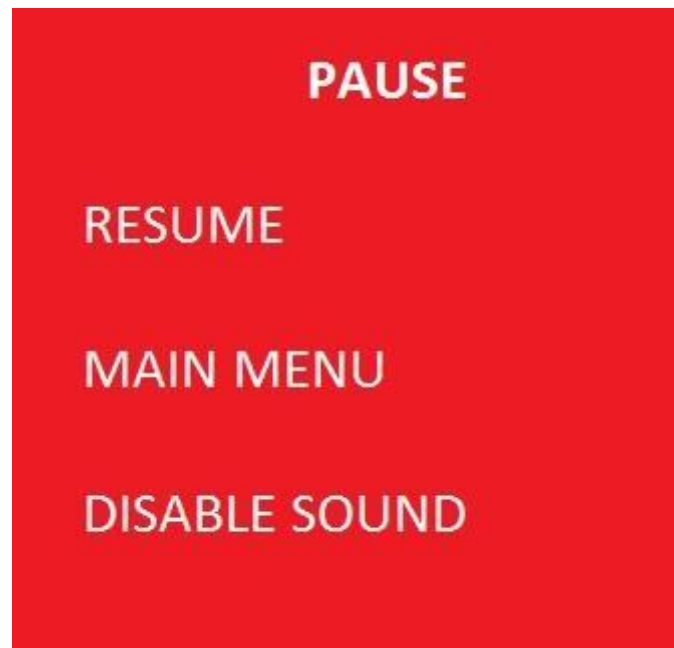-**Enable/Disable Sound:** This option enables or disables the sound when player clicks on to it.

Figure 21: Screen shot of Pause Menu

5.4.2.3 Images Used in Game:

-Bomb: The bomb that aliens throw through the spaceship.

Figure 22: Image of bomb used in game.

-Fire: Fire when the bomb hits the spaceship and game becomes over.

Figure 23: Image of fire used in game.

-Spaceship: Spaceship that player can control during the game.

Figure 24: Image of spaceship used in game.

-Bullet: Bullets are thrown from the spaceship through the aliens.

Figure 25: Image of bullet used in game.

-Aliens: Aliens are the block that player must destroy in order to pass that level.

Figure 25: Image of an alien used in game