
# CS 319 - Object-Oriented Software Engineering

# Analysis Report

Sea Adventures

Group 2-G

Alper Kağan Kayalı

Büşra Oğuzoğlu

Kasymbek Tashbaev

Salih Zeki Okur

# Table of Contents

## 1. Description

Sea Adventures is a horizontal scrolling shooter game based on the game called space shooter. The goal of the project is to implement the game using OOP structure. The game will be implemented using JavaFX.

## 2. Overview

The objective of the Sea Adventures is to pass three levels by defeating enemies and bosses with a submarine, level up and defeat the final boss.

There will be different kinds of enemies in the game, they will be differentiated from their looks and attack scheme by the player. At the end of first two levels of the Sea Adventures, the player will face a boss, and at the final level, the player will face the strongest final boss.

During the game, the player can use certain skills according to submarine's level and get some power up's.

The map that will be used in the game is a visual of the ocean. In addition, the game will be horizontal, unlike Space Shooters. Controllers will be WASD for maneuvering, spacebar for shooting and "z,x,c" keys for skills.

Similar Games that we are inspired by:

Space Shooters: http://playzool.com/games/spaceShooter/?o=portrait

### 2.1. Gameplay

The main hero which is a submarine has health and energy. His goal is to reach the end of the level and destroy final boss in the Scenario mode or survive for more time in the Survival mode. There are many types of enemies with different characteristics that will try to damage the submarine either shooting to it or colliding with. The submarine can shoot and as use its skills to destroy enemies, thereby, it earns some experience and score points. It can collect power-ups to restore health and energy. However, if its health goes to zero, the submarine is destroyed and the game ends.


***Figure 1: The Gameplay of Sea Adventures***

## *2.2. Modes*

There are two different modes in Sea Adventures: Scenario and Survival. In Scenario mode, player should complete 3 levels: in the first to level, player should reach the end of level by destroying enemies in the path and at the end destroy bosses, while in the final level (s)he meets only one enemy, the final boss, which is much stronger than other enemies, and should destroy it to win the game. In Survival mode, there is no end of the level, so the player is going to destroy as many enemies as (s)he can to get more points and survive for more time.

## *2.3. Leveling*

A submarine has a level, which is initially 1 when the game is started. Each time it destroys the enemy, it earns some experience and when it has enough experience, it levels up. However, there is a limit of leveling, so submarine may have at most level 5, and each leveling up will require more experience than the previous one. When submarine's level increases, his health points increases and its weapon and skills become much stronger.

## *2.4. Weapons*

The submarine has a weapon to shoot enemies. As stated in above, when submarine levels up, it improves submarine's weapons, which directly affects the killing efficiency of the submarine. Submarine's attack speed and the bullet's velocity is constant and does not depend on level, however, each leveling up will increase the number bullets submarine releases at each shoot and the damage that bullet deals to enemies.

***Figure 2: Submarine visual of Sea Adventures (from:www.gamedeveloperstudio.com)***

## *2.5. Skills*

The submarine has 3 skills: Mass Destruction, Speed Booster Battery, and System Shield. They are locked at the beginning of the game, however, each leveling up will unlock some skills which will help submarine to kill more enemies or to survive. The skills require energy to be used and they have a cooldown time, which means that the submarine will be unable to use the skills infinitely many times, instead, it needs to wait some time. However, the cooldown time will decrease and their effect will be stronger at each leveling up.

***Figure 3: Mass Destruction Skill visual of Sea Adventures***

## *2.6. Enemies*

Enemies will try to destroy submarine in different ways. They are classified into three group according to their level of difficulty: 1) small enemies, 2) big enemies, 3) bosses. Small enemies have no ability to shoot, so they deal damage only when colliding with the submarine. Big enemies have collision damage as well as ability to shoot. Bosses have high health, shoot multiple bullets each time, and have the ability that releases fireball towards submarine that deals massive damage upon collision. However, they appear only once at each level of Scenario mode and very rare in Survival mode.

***Figure 4: Example visuals of one of the enemies and bosses in Sea Adventures***

## *2.7. Power-Ups:*

There will be power-ups randomly located on the game map to help the player during gameplay. There are two types of power-ups: Repair kit and Fuel. They have different benefits for the player: Repair kit regenerates some health of the submarine, while Fuel restores some amount of energy, so the submarine can use skills more frequently.

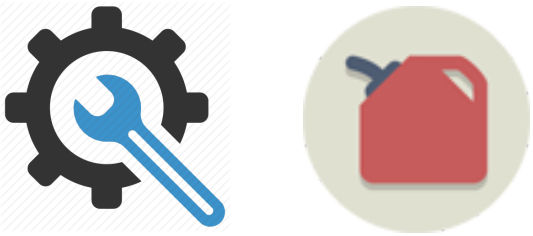


***Figure 4: Visual of Repair Kit and Fuel in Sea Adventures***

# 3. Requirement Specificaiton

## 3.1. Functional Requirements

### 3.1.1. Additional Functional Requirements

**Survival Mode:**

Survival mode will be an alternative way of playing the game. The user will be able to play in survival mode by clicking the button “Play Survival” in the main menu. There are no different map levels in survival mode like there is in regular mode, which is Scenario. Also, Survival mode won’t end by killing a boss. In survival mode the game will continue as long as the submarine is alive. Enemies will come infinitely and strength and quantity of enemies will increase as the game progresses. The goal of the game will be to continue playing the game as long as possible and earn as many score points as possible before dying.

### 3.1.2. Play Game

While playing the game, the user can control only the submarine. The user will have control over its maneuvering, shooting, and skills. Controllers are:

-WASD for maneuvering.

-spacebar for shooting.

-“z,x,c” keys for skills.

By using these three features, the user will try to avoid weapons of little enemies and finally a boss at the end of every level. Enemies will try to shoot or hit the submarine and once they do user will lose some amount of its health according to the strength of the enemy. When submarine shoots an enemy, it will die if it is a small enemy and it will lose health if it is a boss.

The user will be able to start higher levels once he/she killed the boss in a level. Enemies and bosses will get stronger at every level.

### *3.1.3. High Scores*

In the main menu of the game, there will be a high scores button to reach high scores page. From high scores page, the player can reach highest 10 scores that have been made in the game. Player will get points by killing enemies in the game. There will be several different kinds of enemies that will give the player different amount of points based on their difficulty level.

### *3.1.4. Pause Game*

The game can be paused at any time user desires. In addition, the user can continue to the game at any time (s)he wants at the exact location.

### *3.1.5. View Help*

Help page can be viewed by the player from the main menu. From help page, the player can learn about the controls and gameplay of the game.

### *3.1.6. View Credits*

The user will be able to see information about developers of the game from the main menu. There is no credits scene at the end of the final round to not bother the user during game.

### *3.1.7. Change Settings*

The user will be able to enable or disable music and game sounds and change music or game controls from change settings menu which could be reached from the main menu at the beginning of the game or during the game in the pause menu.

## *3.2. Non-Functional Requirements*

### *3.2.1. Game Performance*

Threads will be used to have better game performance. Basically, the game structure will be run at different thread than the GUI part of the game in order to speed up the game.

### *3.2.2. Graphical Smoothness*

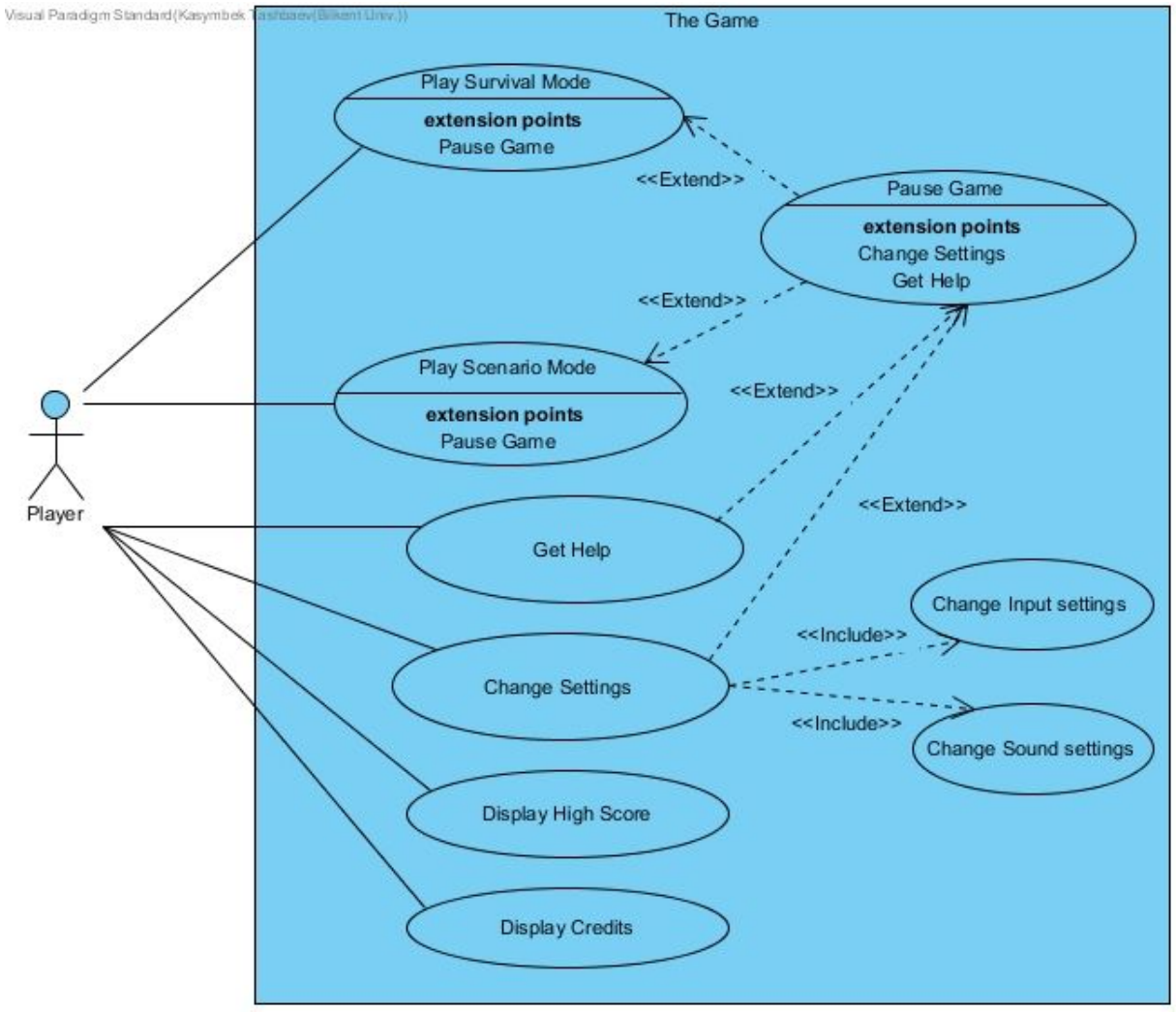
BufferStrategy object will be used to construct interface more smoothly.
Different submarine pictures will be used to make animation.

### *3.2.3. User-Friendly Interface*

The game will be easy to use. First of all, in the main menu, each section will be clickable. Therefore, the main menu will be controlled by just using the mouse. In addition, the main menu will be controllable by arrow keys as well, which will help the game become user-friendly. In addition, the game will be controlled by the keys which are used in almost any games out in the market, which means that the game will be easier to be played by the user since (s)he will be familiar with the controller if (s)he played any game.

# 4. System Model

## 4.1. Use Case Model




### 4.1.1. Play Scenario Mode

**Use Case Name: Play Scenario Mode**

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player aims to complete all levels of Scenario and have the highest score.

-System starts the game and keeps the score of the Player.

**Pre-condition:** Player should be in the main menu

**Post-condition:** No

**Entry Condition:** Player selects “Play Scenraio Game” button from Main Menu.

**Exit Condition:** Player selects “Return to Main Menu” from Pause Menu.

Player looses the game.

Player completes all levels.

**Success Scenario Event Flow:**

1.Game is started by System.

2.Player starts playing from the first level.

3.System creates Enemies on the map.

4.Player moves using the keyboard and reaches the end

5.System creates Boss on the map

6.Player kills the Boss and completes the current level.

7.System grants access to next level.

8.Player starts playing next level.

*Player repeats the steps 3 – 8 until all levels are completed or Player is destroyed.*

9.System record Player’s score to High Score List, if Player score is higher than the lowest score in High Score List and displays the High Score List.

10.System returns to Main Menu.

**Alternative Flows:**

3A. Player shoots the enemy and destroys it:

3A.1. Player presses key corresponding for shooting.

3A.2. System creates bullets forward of Submarine.

3A.3. If the bullet collides with the enemy, its health is reduced by the damage of the bullet.

3A.4. If enemy’s health point is lower or equal to 0, System removes enemy from the map.

3A.5. Player gains the experience according to the destroyed enemy type. If there is enough experience to level up, the level of the Submarine is increased by one, so its weapon is upgraded, and some skills are unlocked.

3A.6. System updates the Player's score according to the type of destroyed enemy.

*Steps 3A.1 – 3A.6 are applied everytime Player shoots.*

3B. Player collects the power-ups during the game:

3B.1. System creates Power-up.

3B.2. Player moves towards to the power-up.

3B.3. Player collects power-up by colliding with it.

3B.4. The system removes the power-up and makes necessary changes to apply the effect of power up.

3B.5. System updates the Player's score according to the collected power-up.

*Steps 3B.1 – 3B.6 are applied everytime Power-up appears.*

3C. Player takes damage and is destroyed:

3C.1. Player collides with Enemy or its bullet .

3C.2. Player's health point is reduced by the collision damage or damage of the bullet.

3C.3. If its health point of Player is lower or equal to 0, System ends the game and returns to the Main menu.

3D. Player uses skill:

3D.1. Player presses key corresponding for some skill.

3D.2. Player's energy decreased by energy cost of the skill.

3D.3. System make skill unavailable and launches cooldown time.

3D.4. System apllies the effects of the skill.

*Steps 3D.1 – 3D.4 are applied everytime Player uses skill.*

A. Player requests to pause the game at any time during the game:

A.1. "Pause Game" use case is applied

### *4.1.2. Play Survival Mode*

**Use Case Name: Play Survival Game**

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player tries to survive on the wave of enemies and get high score points.

-System keeps the score of the Player.

**Pre-condition:** Player should be in the main menu

**Post-condition:** No

**Entry Condition:** Player selects “Play Survival” button from Main Menu.

**Exit Condition:** Player selects “Return to Main Menu” from Pause Menu.

Player looses the game.

**Entry Condition:** Player selects “Play Survival Game” button from Main Menu.

**Exit Condition:** Player selects “Return to Main Menu” from Pause Menu.

**Success Scenario Event Flow:**

1.Game is started by System.

2.Player starts playing the game.

3.Player moves, shoots and uses skill using keys.

4.System updates score, when enemy is destroyed.

*Player repeats the step 2-4 until Player is destroyed.*

5.System records Player’s score to High Score List, if Player score is higher than the lowest score in High Score List, and displays High Score List.

6.System returns to Main Menu.

*Player repeats the steps 1 – 4 if he wants to play game again.*

**Alternative Flows:**

3A. Player collects the power ups during game:

3A.1. System creates Power-up.

3A.2. Player moves towards to the power-up.

3A.3. Player collects power-up by colliding with it.

3A.4. The system removes the power-up and makes necessary changes to apply the effect of power up.

3A.5. System updates the Player's score according to the collected power-up.

*Whenever a power-up appears during the game, steps 3.A.1-3.A.5 are applied.*

A. Player requests to pause the game at any time during the game:

A.1. "Pause Game" use case is applied

### *4.1.3. Pause Game*

**Use Case Name: Pause Game**

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player wants to pause the game during the game.

-System displays Pause menu.

**Pre-condition:** Player should be playing the game

**Post-condition:** No

**Entry Condition:** Player presses the proper key from keyboard to Pause the game.

**Exit Condition:** Player resumes the game selecting "Return to the Game" from the Pause menu

Player presses “Ecs” key.

Player selects “Exit Game” from the Pause menu.

Player select “Return to Main menu” from the Pause menu

**Success Scenario Event Flow:**

1. Player presses the proper key from keyboard to pause the game.
2. System pauses the game.
3. System shows the pause menu.

**Alternative Flows:**

3A. Player wants to resume the game:

3A.1.Player selects “Return to the Game” from the pause menu or presses “Esc” key.

3A.2.System resumes the game.

3B. Player wants to exit the game:

3B.1.Player selects “Exit Game”

3B.2.System closes the game and returns to desktop.

3C. Player return to the Main menu;

3B.1.Player selects “Return to Main menu”

3B.2.System closes the game and returns to Main menu.

### *4.1.4. Display Credits*

**Use Case Name: Display Credits**

**Primary Actor**: Player

**Stakeholders and Interests**:

- Player wants to learn the names of the developers
- System displays the names of the developers.

**Pre-conditions**: Player should be in Main Menu.

**Post-condition**: No

**Entry Condition**: Player selects “View Credits” from the main menu.

**Exit Condition**: Player selects “Back” to return the previous menu.

**Success Scenario Event Flow:**

1.System shows the name of the developers of the game Sea Adventures.

**Alternative Flows:**

A. If player desires to return the main menu at any time:

A.1. Player selects “Return to Main Menu” button to return main menu.

A.2. System displays Main Menu.

### *4.1.5. Display High Scores*

**Use Case Name: Display High Scores**

**Primary Actor**: Player

**Stakeholders and Interests:**

-Player wants to see top ten scores with player names.

-System shows the list containing top ten scores with player names.

**Pre-conditions:** System keeps records of top ten scores.

Player should be in the Main Menu

**Post-condition:** No

**Entry Condition:** Player selects “View High Scores” from Main Menu.

**Exit Condition:** Player selects “Back” to return Main Menu.

**Success Scenario Event Flow:**

1.System displays top ten scores with player names.

**Alternative Flows:**

A. If player desires to return main menu at any time:

A.1. Player selects “Return to Main Menu” button to return main menu.

A.2. System displays Main Menu.

### *4.1.6. Get Help*

**Use Case Name: Get Help**

**Primary Actor:** Player

**Stakeholders and Interests:**

- Player wants to learn about the gameplay and controls.
- System shows basic principles of the game, the objectives and the controls.

**Pre-conditions:** Player should be in Main menu.

Player should pause the game and be in Pause menu.

**Post-condition:** No

**Entry Condition:** Player selects “View Help” from the Main menu.

Player selects “View Help” frim the Pause menu.

**Exit Condition:** Player selects “Back”.

**Success Scenario:**

1.Player selects “View Help” from the menu (Main or Pause).

2. System shows the objectives of the game and controls on a separate page.

**Alternative Flows:**

B. Player requests to return to the previous menu at any time:

B.1. Player selects “Back” button from “View Help” screen.

B.2. Player returns to the previous menu (Main or Pause menu).

### *4.1.7. Change Settings*

**Use Case Name: Change Settings**

**Primary Actor**: Player

**Stakeholders and Interests**:

-Player wishes to change game settings: changing the volume of the game and the background music or changing the keys inputs.

-System saves the settings which are changed by the player.

**Pre-condition:** Player should be in Main menu.

Player should pause the game and be in Pause menu.

**Post-condition:** Sound settings are updated.

Input settings are updated.

**Entry Condition:** Player selects “Change Settings” button from the Main menu.

Player selects “Change Settings” button from the Pause menu.

**Exit Condition:** Player selects “Back” to return menu.

**Success Scenario Event Flow:**

1. Player presses “Change Settings” button to make changes to the game settings.
2. Game settings are displayed to Player in “Change Settings” screen by System.
3. Player adjusts settings according to his desire.

4. System updates game settings successfully.

**Alternative Flows:**

A. Player desires to use default settings at any time:

A.1. Player selects “Default Settings” button from “Change Settings” screen.

A.2. The system changes settings as default.

B. Player requests to return the previous menu at any time:

B.1. Player selects “Back” button from “Change Settings” screen.

B.2. The system asks whether Player wants to save his/her changes or not.

B.3. Player selects “Yes” or “No” whether to save or not to save the settings.

B.3. Player returns to the previous menu (Main or Pause menu).

## *4.2. Dynamic Models*

### *4.2.1. Start Game*

Following sequence Diagram illustrates the scenario explained below:

**Scenario:** Player Uğur presses start game button from the main menu of the game. After pressing the button system initializes the map according to which level player is playing at that current time. The system will also initialize objects such as the submarine and the enemies and put them in their first location. After that game loop will start and continuously update the map and the locations of the objects and the score according to player’s performance.

OBJECTS IN GAME

- GameEngine
- SoundEngine
- CollisionManager
- GameObject
- SmallEnemy

- BigEnemy
- Boss
- Submarine
- Skill
- SkillManager
- MapHealth
- Energy
- Cooldown Manager
- PowerUp
- Object Random Location Manager
- SettingManager
- InputManager
- ScreenManager
- MainMenu
- PauseMenu

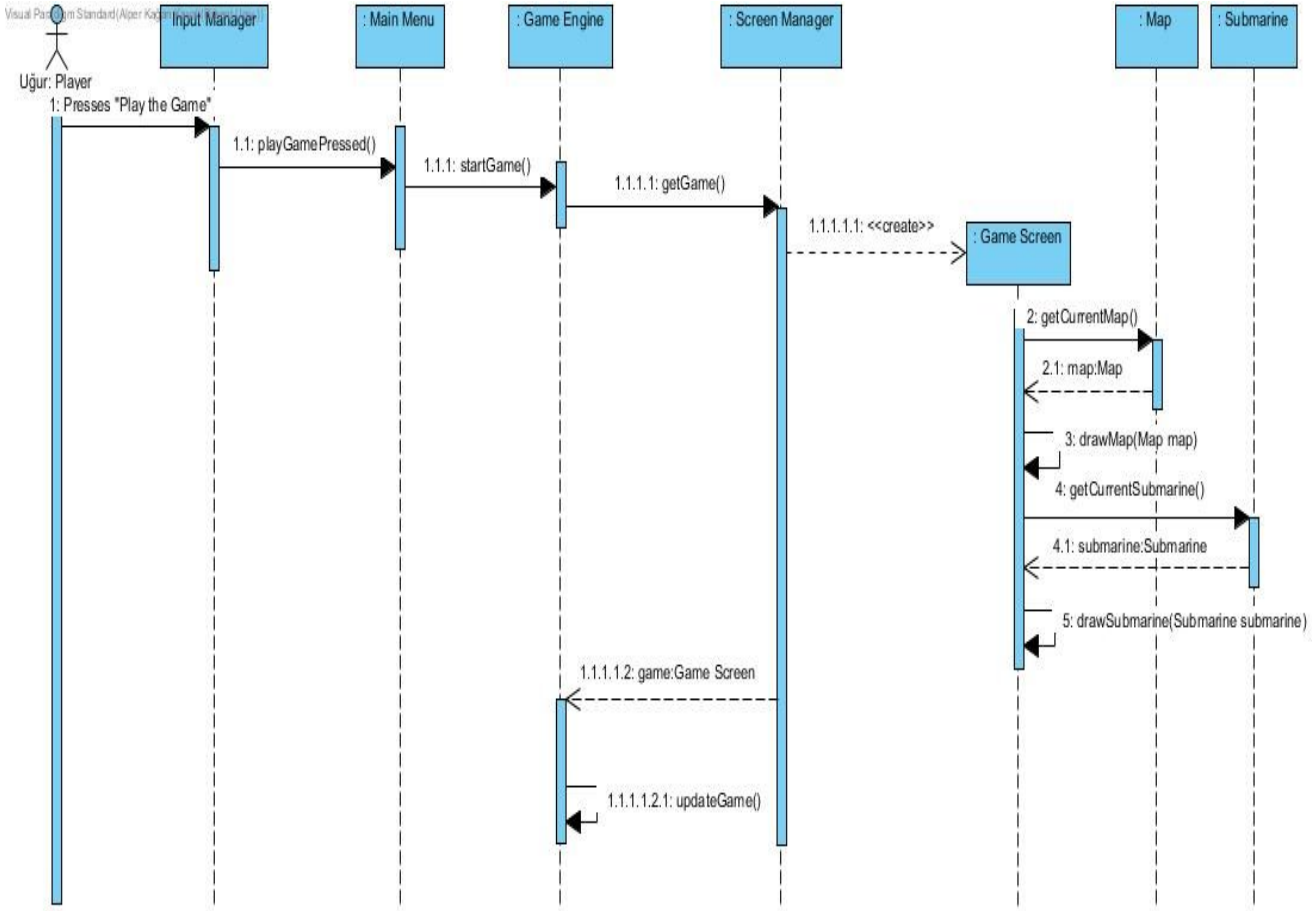


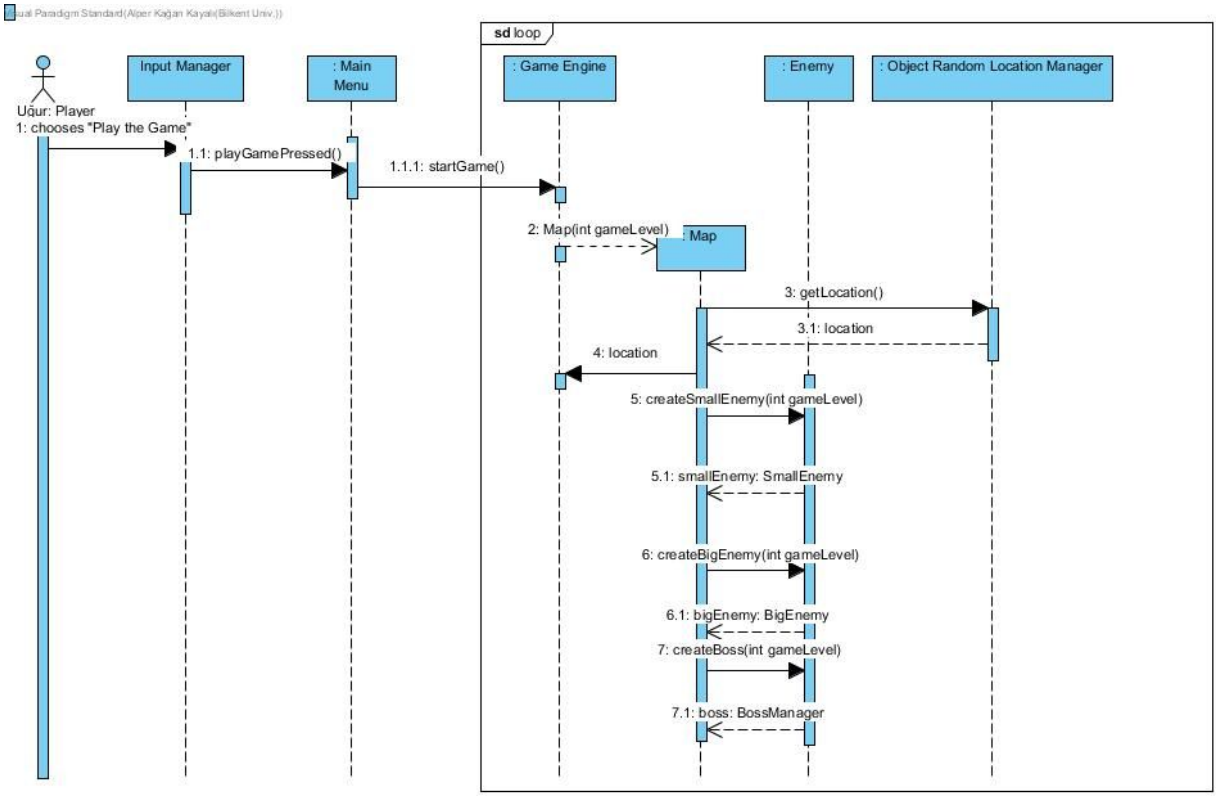

**Figure 4.2.1.1 shows the sequence diagram which explains start game scenario**

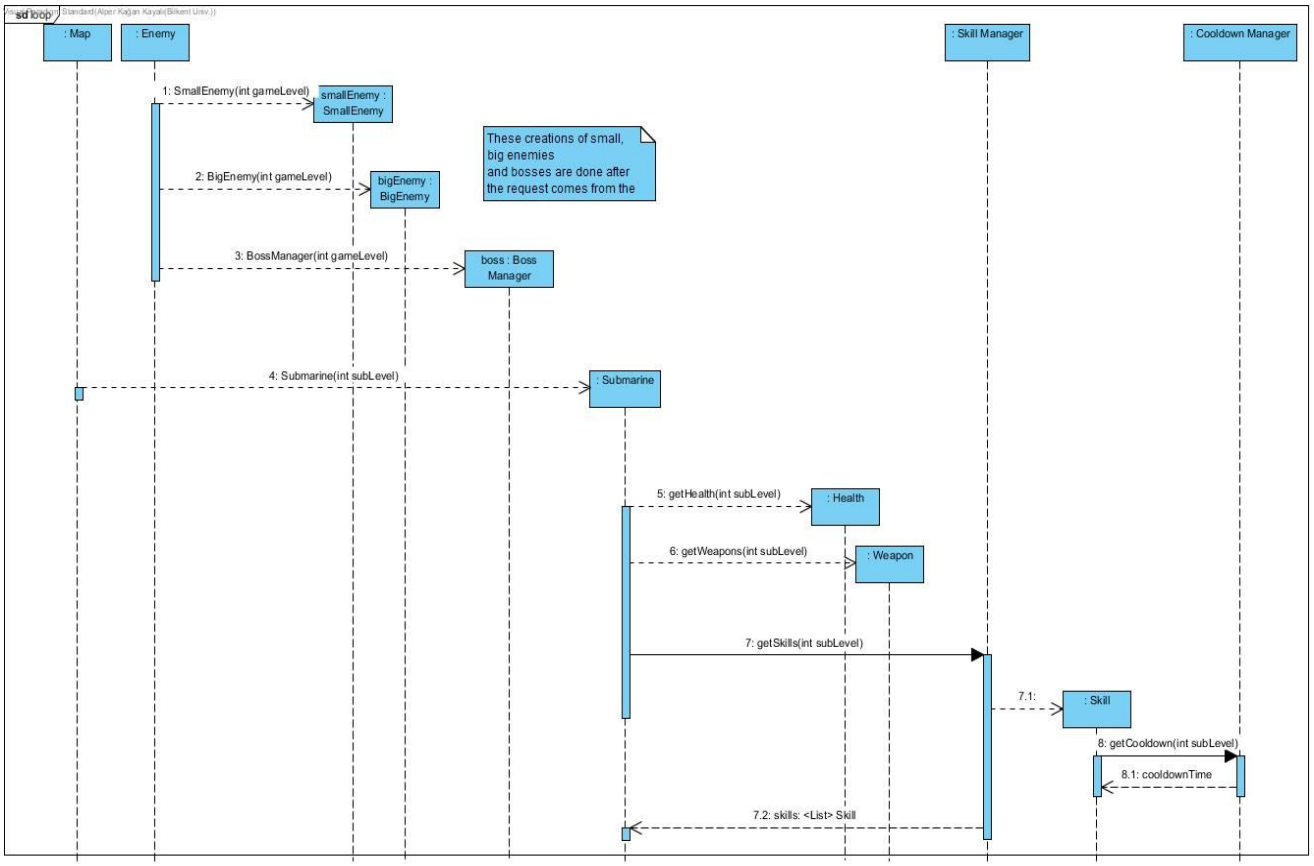
Here MainMenu is a boundary object by which user can perform some activities like starting the game, viewing credits, changing settings. In addition, ScreenManager handles creating full-screen window and graphics issues. The locations of the game objects are stored in GameMap class, GameMapManager is responsible for the organization of the GameMap. Hence, System enters a game loop in order to continuously update the game.

### *4.2.2. Playing the Game*

Following sequence Diagram illustrates the scenario explained below:

**Scenario:** Player Uğur starts the game by pressing the button from main menu and game starts as it is explained in the section above. In game loop, system checks for any movement request from the player by using specified buttons and updates the location of submarine accordingly. At the same time, it checks if any collision happened with the enemy ships or either side shoot each other. According to these, pictures will change in order to make an animation. The system will check the locations and compare them to achieving these results. If the player managed to shoot enemies, the score will get updated accordingly. If the player managed to kill the boss at the end of the map, a new map will be loaded by the system and objects will be placed as described in the section above.




sd loop
: Map
: Enemy
: Skill Manager
: Cooldown Manager
1: SmallEnemy(int gameLevel)
smallEnemy : SmallEnemy
These creations of small, big enemies and bosses are done after the request comes from the
2: BigEnemy(int gameLevel)
bigEnemy : BigEnemy
3: BossManager(int gameLevel)
boss : Boss Manager
4: Submarine(int subLevel)
: Submarine
5: getHealth(int subLevel)
: Health
6: getWeapons(int subLevel)
: Weapon
7: getSkills(int subLevel)
7.1:
: Skill
8: getCooldown(int subLevel)
8.1: cooldownTime
7.2: skills: <List> Skill

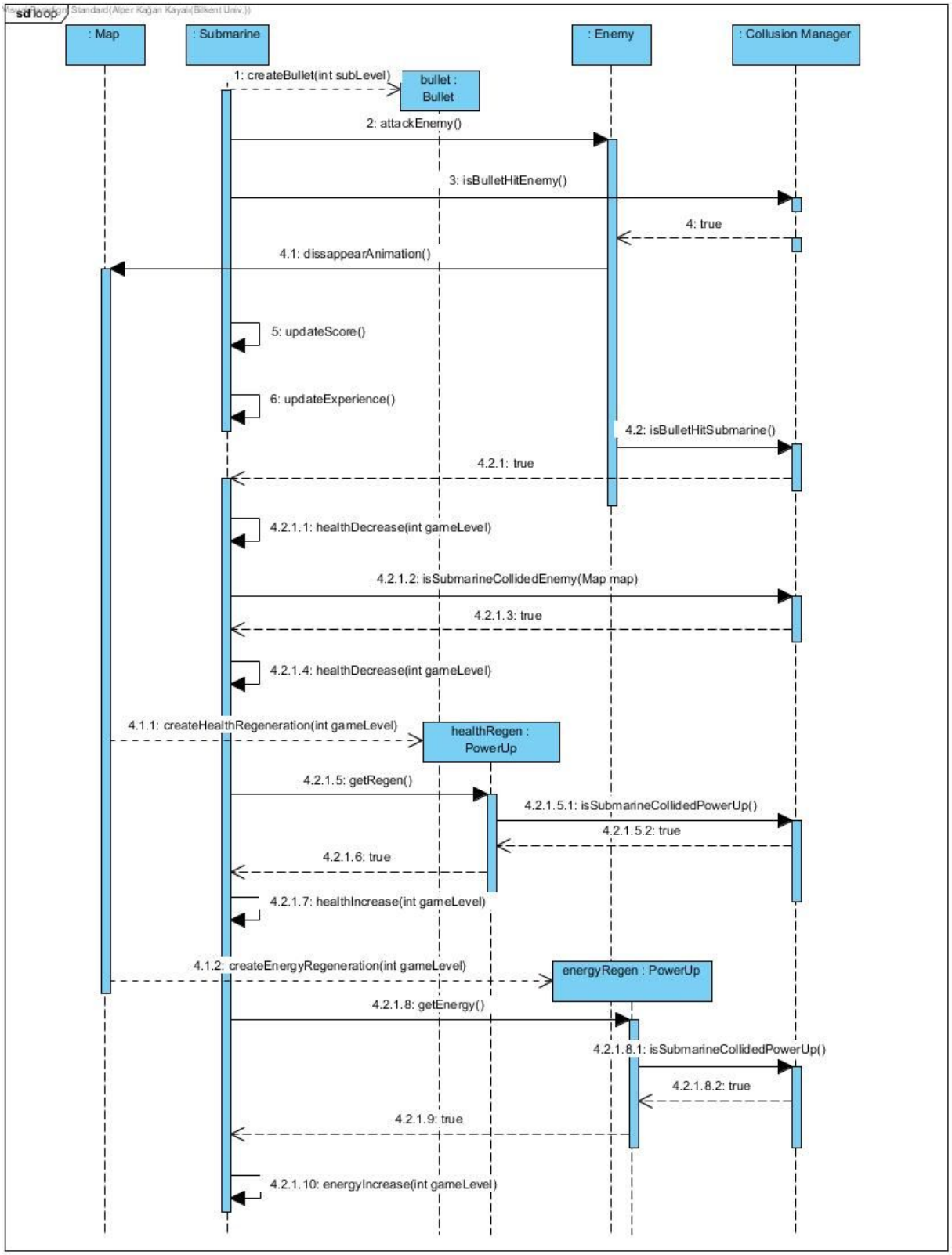



**Figure-4.2.1.2 shows the sequence diagram which explains Gameplay**

Here is another way to look at the gameplay:

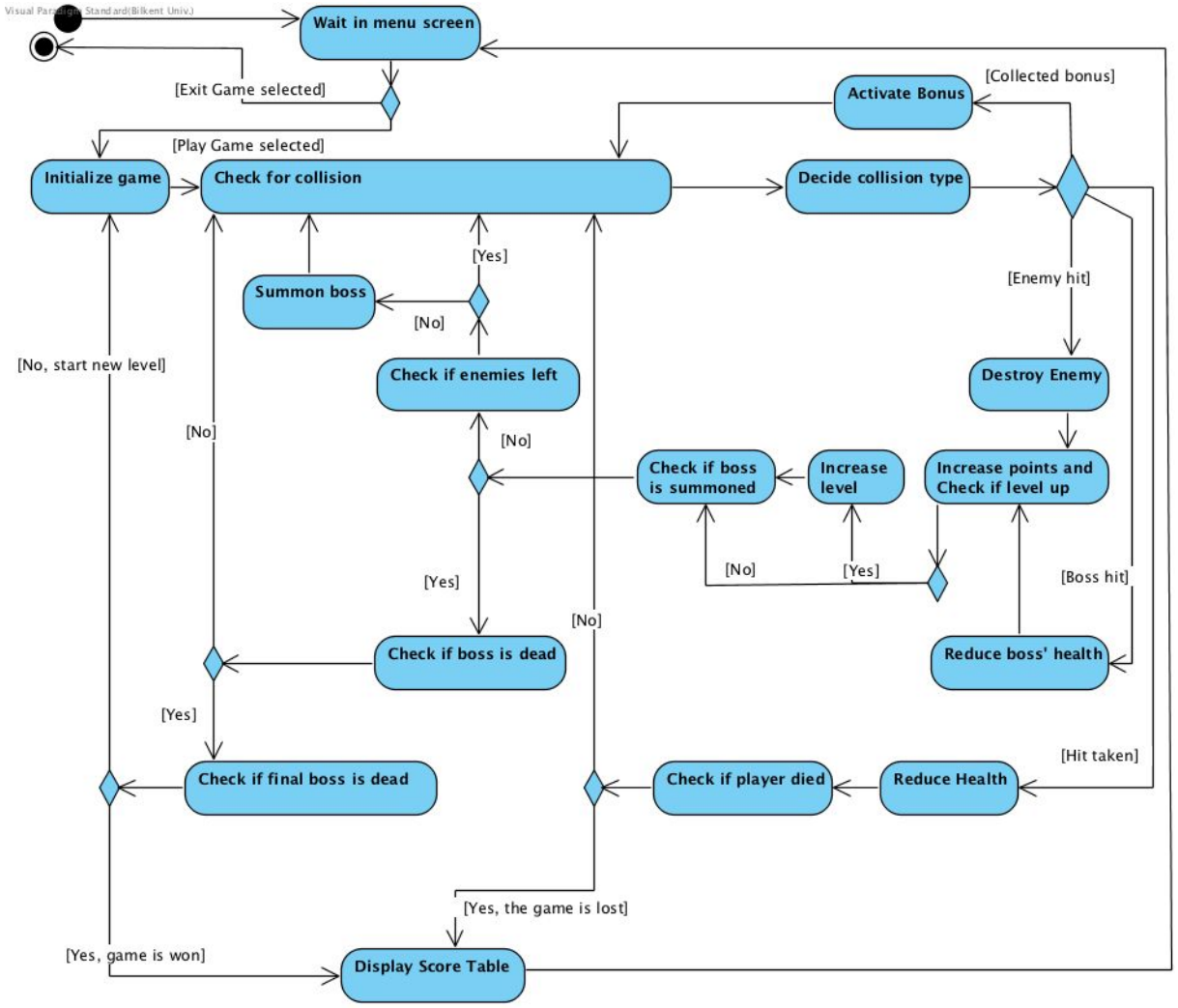



**Figure 4.2.2 Illustrates Activity Diagram of in-game activities**

When the user selects “Play Scenario Game” button from main menu system initializes game, instantiates the game object and creates the first game map. When this is finished, the game starts. Enemies start to attack the player. After that, the player starts to attack the enemies and system checks if there are any collisions or not. If the player hits enemies, points section will be updated. When the player collects a certain amount of points, the system will update submarines level which will boost submarines stats. If submarine gets hit a certain amount of time, the game will end by player losing. After beating a certain amount of enemies, the boss will appear. Player will start to attack the boss and system will constantly check for collisions and update boss’s health accordingly. When the player defeats the first boss, the game will proceed to the second level and the system will initialize the second map and new enemy

objects as well as second boss object. The same process will repeat until player beats the second boss. At the third level after beating the final boss, the game will end by player winning and score of player will be shown. Then the game will proceed to the main menu. The system will check if player's score is in top ten high scores or not. If it is, high scores page will update accordingly and the player can start the game again if (s)he wants to.

### *4.2.3. Change Settings*

Following sequence diagram illustrates the scenario explained below:

**Scenario:**Player Uğur presses the Change Settings button from Main Menu. The System displays current settings and buttons to change the settings. Player changes the volume of in-game sound effects, music, and first action key. Then, player presses back to the menu button, System applies new settings to Game Engine and asks Uğur if he wants to save the current settings or not. Uğur presses no button. After that Main Menu displays the main menu.

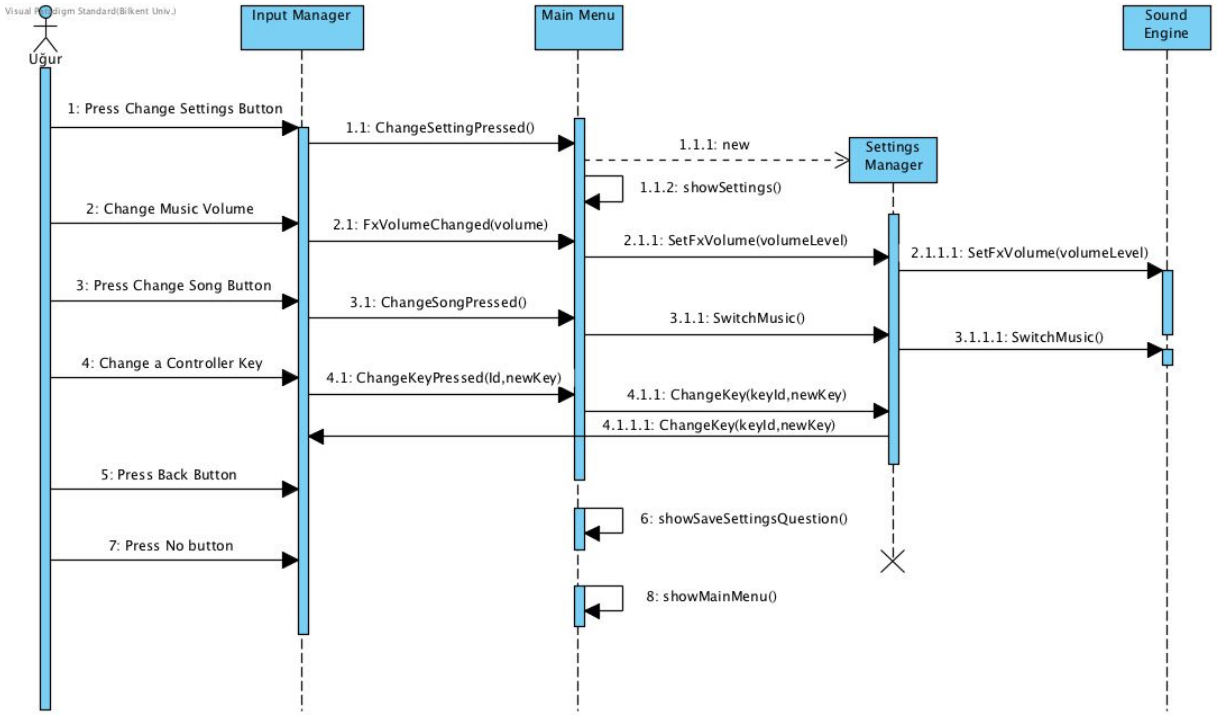
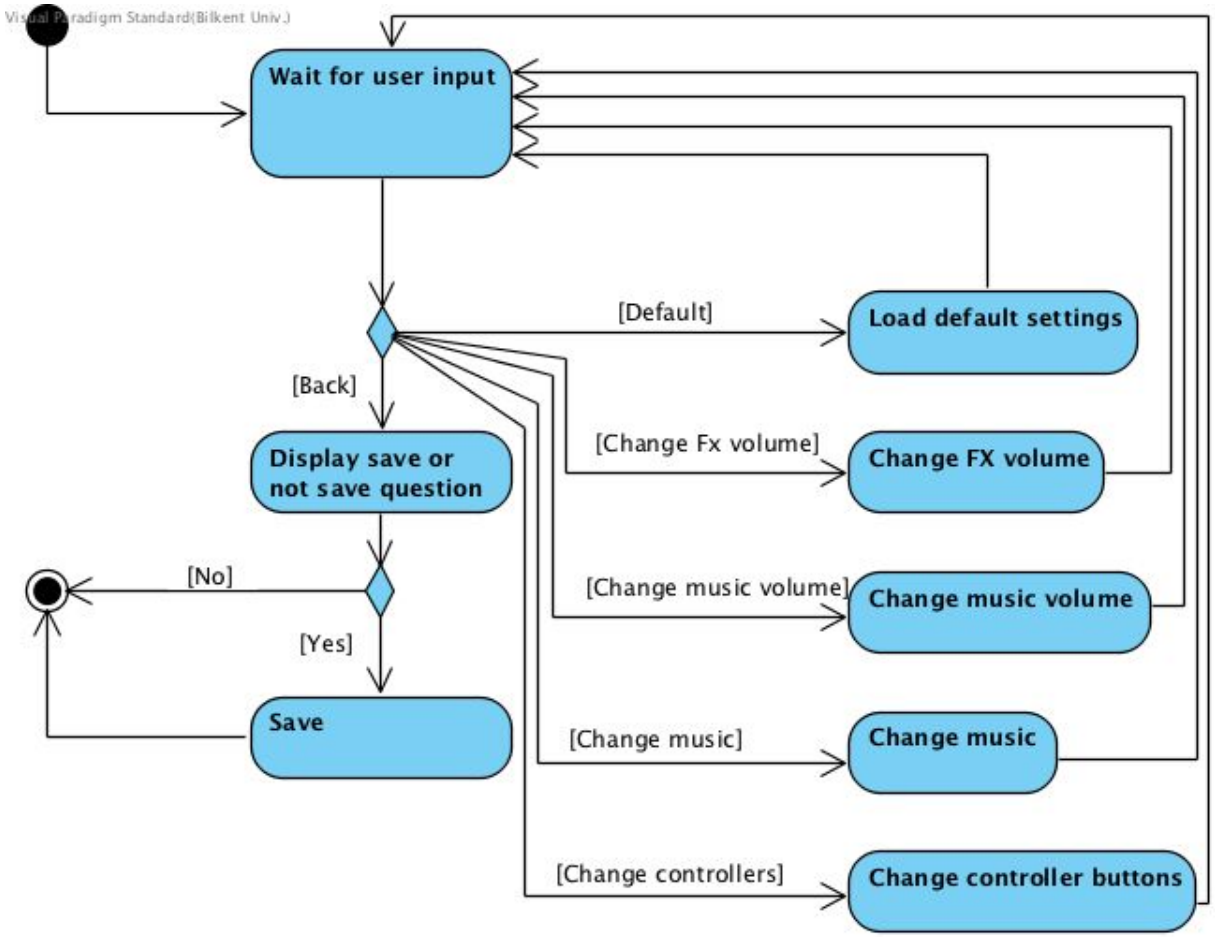



**Figure-4.2.1.2 illustrates the change settings sequence diagram**

Here, Game Engine sends the name of music to be opened and desired music volume degree to Sound Engine. Game Engine also calls its own method with Id of keyboard key which will be replaced with previous action key.

Following diagram shows how system maintains a change in settings.




When the user clicks to change settings in the main menu, setting screen appears. User can change music, controller keys, music volume, the volume of in-game sound effects and load default settings or return to the main menu from settings screen. When the user presses back button a new panel opens that asks the user if he/she wants to save current settings. If user chooses to save current settings, they are saved as default settings. After that main menu screen appears.

## *Object and Class Model*

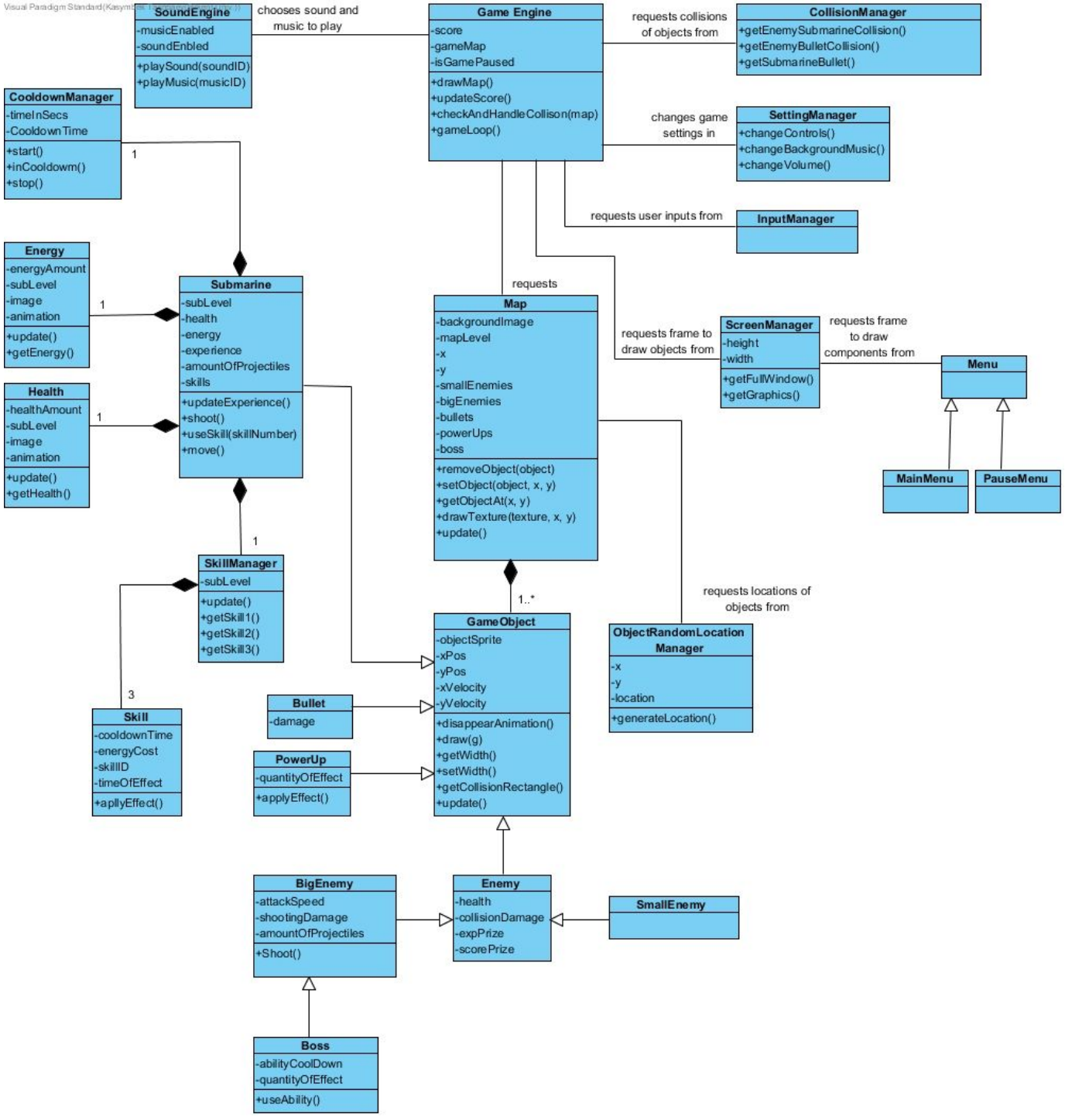


**Figure-4.3 Class diagram of Sea Adventures game**

Object model of the Sea Adventures game is illustrated in above diagram.

*Brief explanations about classes are given in below section.*

- **GameEngine** manages the game loop and relations between other objects
- **SoundEngine** plays desired sound or music, if they are enabled in the settings
- **CollisionManager** checks whether there is a collision between submarine, enemies and bullets
- **GameObject** parent class of all game objects that holds their information
- **Enemy( SmallEnemy, BigEnemy, Boss)** holds information about enemy's damage, health, velocity and location
- **Submarine** holds information about submarine's stats, damage, skills, velocity and location
- **Skill** holds information about skill's effect, energy cost, cooldown
- **SkillManager** all operations on skills performed via this class
- **Map** holds information about current objects in the map
- **Health** holds information about submarine's health and draws and updates health bar according to this information
- **Energy** holds information about submarine's energy and draws and updates energy bar according to this information
- **Cooldown Manager** holds information about skills' cooldownds and draws and updates skill bar according to this information
- **PowerUp** holds information about powerup's effect and its quantity
- **Object Random Location Manager** generates random x and y locations, so map can use to locate objects in random positions
- **SettingManager** changes game settings and submarine control
- **InputManager** when user uses input hardware detects which key was pressed
- **ScreenManager** holds information about resolution of the screen and graphics
- **Menu( MainMenu, PauseMenu)** detects which section of the menu was selected

# 5. User Interface

## 5.1. Navigational Path

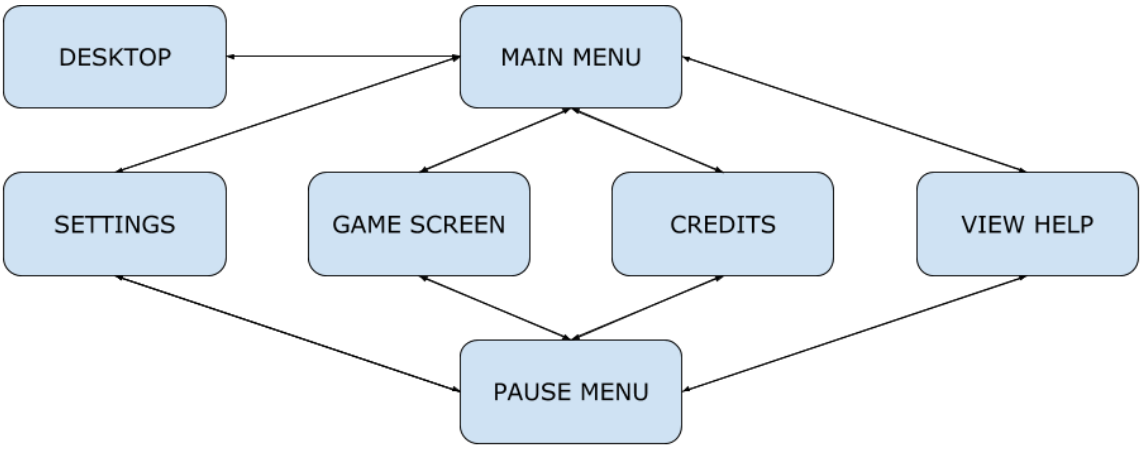



**Figure 5.1 Illustrates the Navigational Path of Sea Adventures**

## 5.2. Screen Mock-ups

### 5.2.1. Main Menu

When game begins to run, main menu will appear. There will be several options in main menu:

- Play Scenario Game
- Play Survival Mode
- View Highscores
- View Help
- Change Settings
- Credits
- Exit Game


**Figure 5.2.1 Screen shot of Main Menu**

**-Play Scenario Game:** When player selects “Play Scenario Game” option from main menu, game starts with current settings from the current level player is in. This mockup both covers  scenario mode and the survival mode.

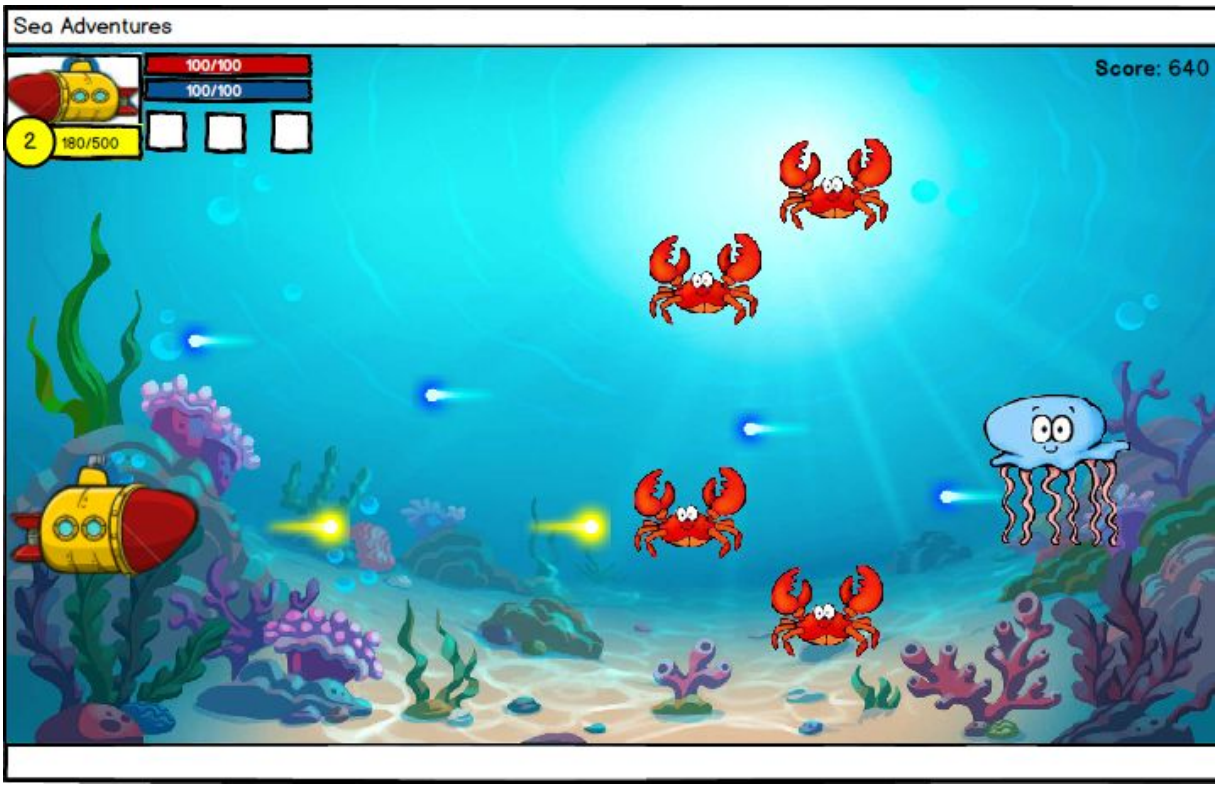
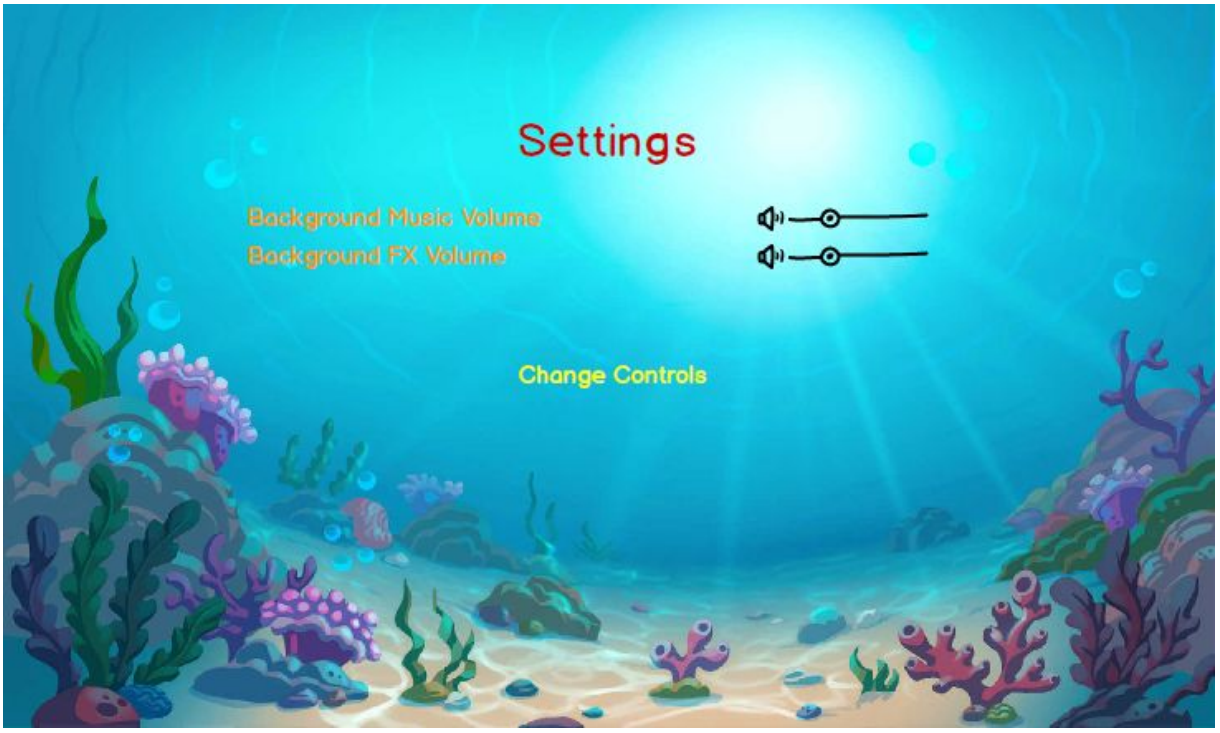



**Figure 5.2.1.1 Screen shots of gameplay**

**-Change Settings:** Change Settings button appears in main screen. When player presses the Change Settings button, settings menu will appear. From the settings menu, options will appear to change the music that has been played during gameplay, the volume of the music, and which controls player is going to use during gameplay.



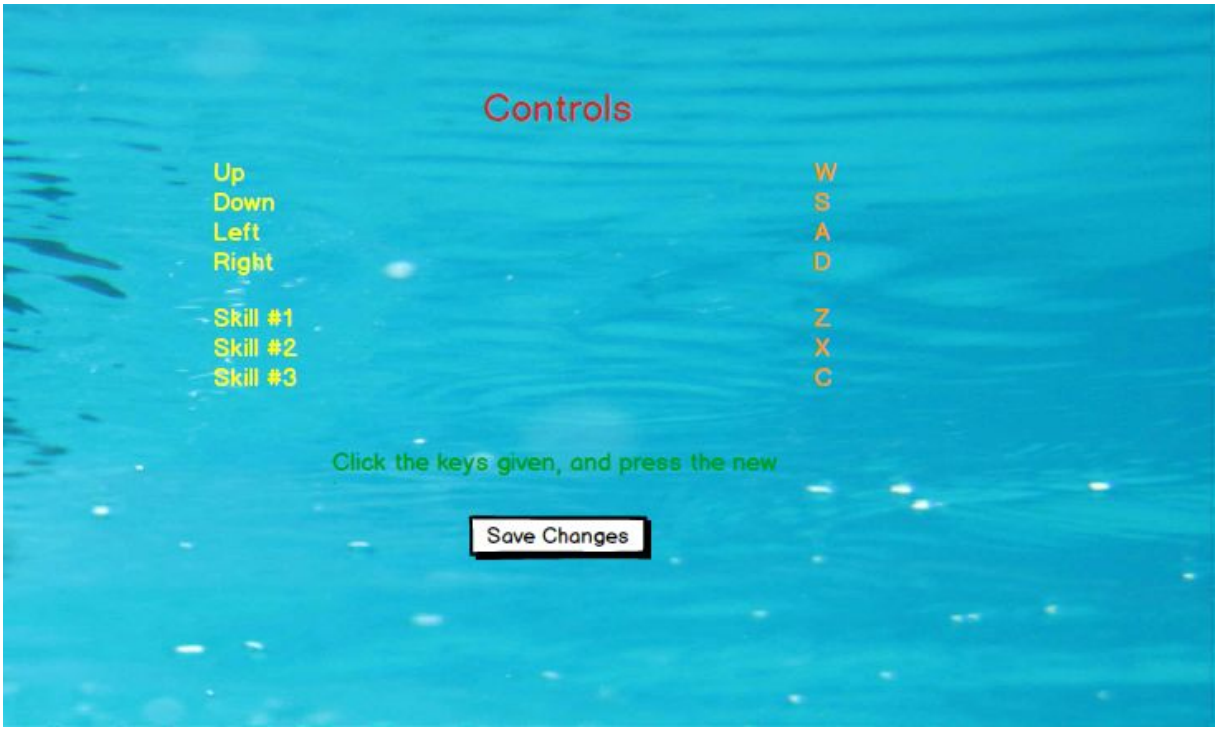



**Figure 5.2.1.2 Screen shot of Change Settings**

**-View High Scores:** When High Scores button pressed from main menu, first 10 highest highscores will appear in following screen. Player can go back to main menu using back button.


**Figure 5.2.1.3 Screen shot of View High Scores**

**- View Help:** If player chooses to view help from main menu, help menu will appear and show information about gameplay and controls.

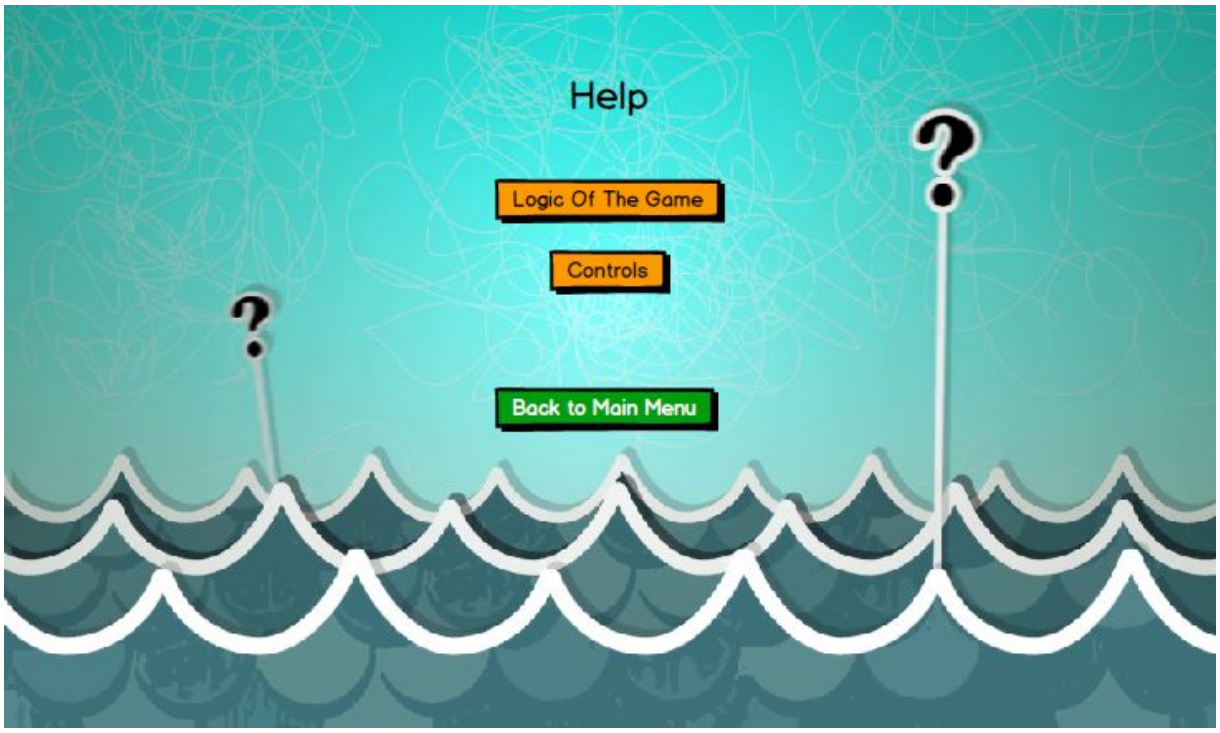



**Figure 5.2.1.4 Screen shot of Help**

**-Credits:** When player wishes to view credits, related page will be appear when player presses the Credits button from main screen. Player can go back to main screen by pressing the Back button.

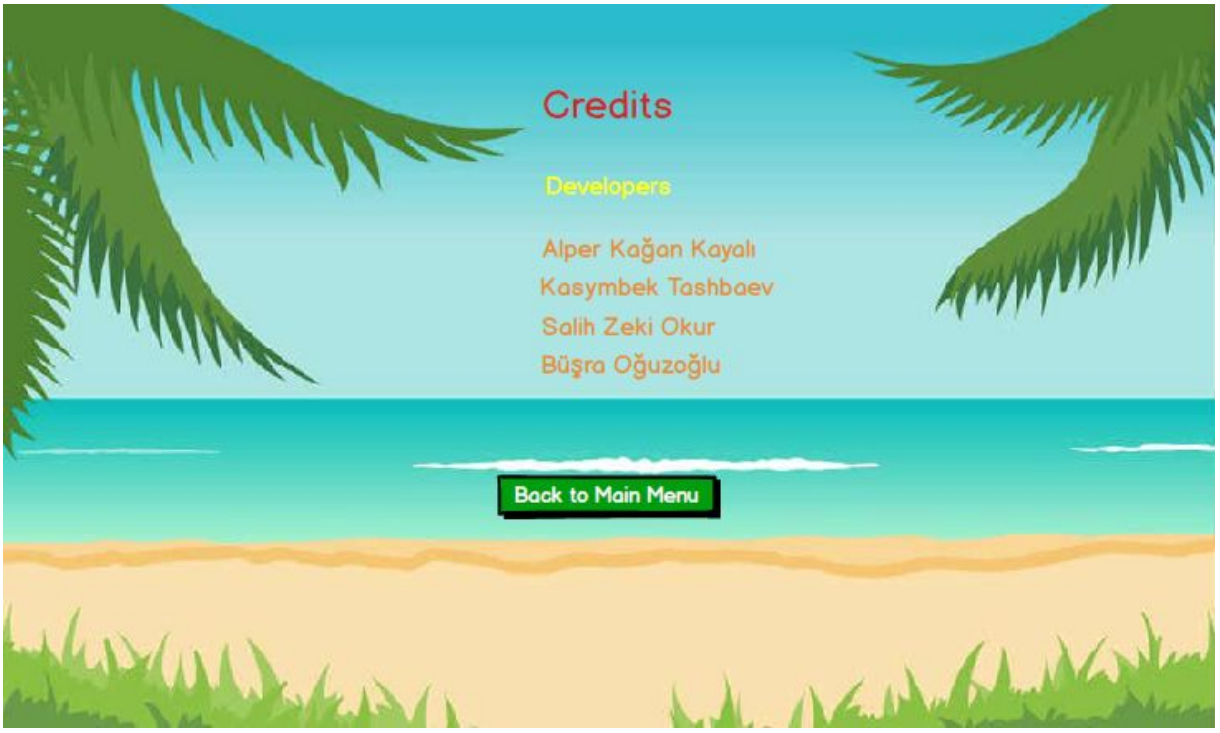



**Figure 5.2.1.5 Screen shot of Credits**

**-Exit Game:** When player wishes to exit the game, can go back to main menu and press Exit button from main menu.

### *5.2.2. Pause Menu*

If the player wants to pause the game, (s)he should press the escape button (ESC). After that, pause menu will be shown on screen. The options of the pause game are similar to the main menu. However, exit game option is not in pause menu. Instead, there is a “Back to Main Menu” option. Therefore, if player wants to exit the game, he should select “Back to Main Menu” then he can exit the game. (Figure 5.2.2)

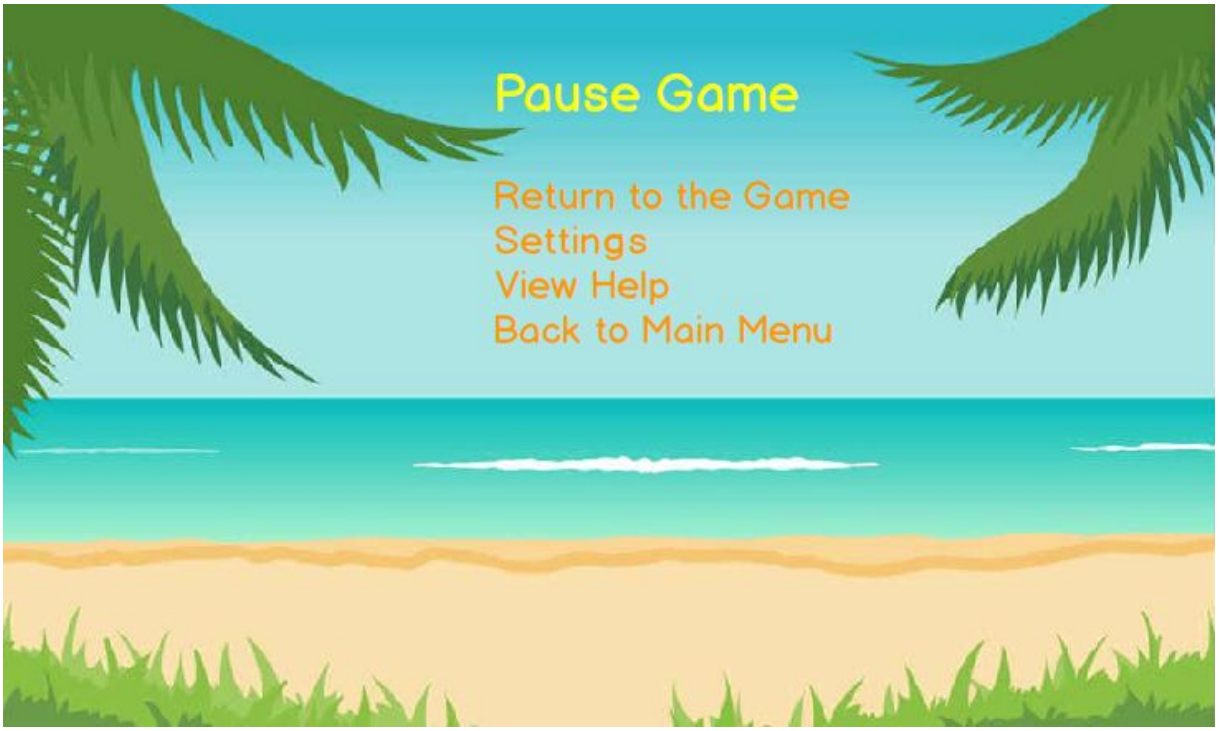
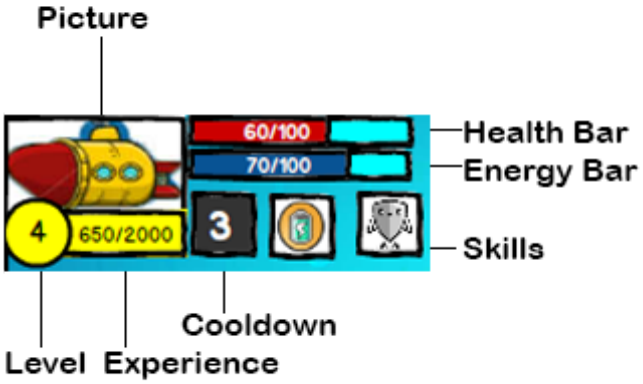



**Figure 5.2.2 Screen shot of Pause Menu**

### *5.2.3. Submarine:*

Submarine in the game:

Submarine’s stats




### *5.2.4. Skills*

**Locked:** Skill cannot be used, until submarine reaches certain level.

**Weapon of mass destruction:** deals massive damage to all enemies in the map.
Level of submarine required to unlock: 2
Energy required: 50
Cooldown: 30 sec (decreases by 3 at each level of submarine)

**Speed booster battery:** Submarine shoots twice faster for 10 sec.
Level of submarine required to unlock: 3
Energy required: 30
Cooldown: 40 sec (decreases by 5 at each level of submarine)

**System shield:** Submarine becomes invulnerable for 5 sec.

Level of submarine required to unlock: 4

Energy required: 40

Cooldown: 30 sec (decreases by 3 at each level of submarine)

### *5.2.5. Power ups:*

**Repair kit:** regenerates health

**Fuel:** restores energy.

### *5.2.6. Enemies*

***Small Enemies:***

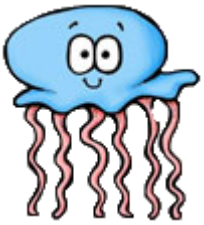
**Happy Crab**: slowly moves towards the submarine, deals medium damage when collides

**Angry Cancer hermit:** very quickly moves towards the submarine, deals medium damage when collides

**Underwater Mine:** very slowly moves towards the submarine, deals massive damage when collides

***Big Enemies:***

**Dark Reef Coral:** slowly moves vertically and horizontally towards submarine, shoots multiple bullets each time at very slow rate; bullets deal little damage



**Pipsqueak jellyfish:** fastly moves vertically and horizontally towards submarine, shoots multiple bullets each time at medium rate; bullets deal high damage

**Seahorse:** slowly moves vertically and horizontally towards submarine, shoots one bullet each time at high rate; bullets deal high damage

***Bosses***

***Boss properties:*** All bosses have high health, shoot multiple bullets each time, and have ability that releases fireball towards submarine that deals massive damage upon collision.

*End map bosses of the first level:*

**Cowardly octopus**

*End map bosses of the second level:*

**Blue whale**

*Final Boss:*

**Kraken**

## 6. *Important Decisions in Overall Analysis*

- In analysis report, we have decided on the functional requirements which are the ones that customer needs, also we decided on the non-functional requirements which should be implemented in order to increase the quality of the software.
- We decided on important features of our game such as leveling system, high scores and settings, and we decide how to implement them.
- We decided and designed the object and class model of our game.
- We made the dynamic model of our game which will help us on the implementation of the game.
- We decided and designed the user interface of our game.

## 7. *Conclusion*

In this analysis report, we designed diagrams and models in order to implement our game Sea Adventures.

Our report consists of two different important design pieces including requirements and system model.

Requirements part consists of functional and non-functional requirements. In this part, we tried to think about all functional and non-functional requirements for a player to play our game. We designed our system model and fill the report according to those requirements.

System model consists of four different sections which are use case model, dynamic model, class model and user interface.

We decided on our use cases by taking our requirements into consideration, made a use case diagram and explained each of our use cases in detail.

Dynamic model consists of sequence diagrams, state diagrams as well as activity diagram of our game. In those parts we tried to clarify how our game will work as a system and how it's interactions will affect that system. We made diagrams and analysis for starting the game, actual gameplay and changing the settings of the game since these are crucial parts of our system. In diagrams, we identified all possible states in the game. In the activity diagram, we tried to clarify how our game system works in possible scenarios. While doing the analysis of the dynamic model, we decided on our objects that we will use in our game and made object and class models according to that.

Fourth part of our analysis report is user interface and navigational path diagram. The user interface basically designed in order to make the game easier to interact. There are 5 options in the main menu of the game which can be easily understood by the player. The first option of the main menu is "Play Scenario Game", which directly creates a new game for the user. The second option is settings, which will help the user to change the settings of the game and the controls. The third option of the game is the credits, which will allow the player to see the developers of the game, the fourth option is view help, in order to help the player to get into the game. Final option is quit game, which will automatically close the game if it is selected. Each help, credits and settings menu has a back to the main menu option, which will allow the user to go to the main menu. Finally, the user can pause the game which will help the user to change the settings and view help while the game is saved at the exact location. Of course, the player can continue the game after the changes are made.

The navigational path shows scenes that can be reached by the user in the game. Which include game scene, settings, view help, credits scenes and main menu, pause menu scenes which can be reached to and reached by previous four scenes. User enters to and exits from the game through the main menu screen.

## *8. References*

-2d torpedo game assets. (n.d.). Retrieved March 30, 2018, from https://www.gamedeveloperstudio.com/graphics/viewgraphic.php?item=1r4390441v055y2r6u

-Animation of tropical landscape - beach, sea,... (n.d.). Retrieved March 30, 2018, from https://www.shutterstock.com/video/clip-6445103-stock-footage-animation-of-tropical-landscape-beach-sea-waves-palms-green-screen-before-and-after.html?src=rel/6426590:1/gg

-Dynamic graphic animation using paper cutout... (n.d.). Retrieved March 30, 2018, from https://www.shutterstock.com/video/clip-1206676-stock-footage-dynamic-graphic-animation-using-paper-cutout-styled-elements-to-illustrate-a-sea-of-questions-high.html?src=rel/1206664:4

-Bruegge, B., & Dutoit, A. H. (2014). Object-oriented software engineering: Using UML, Patterns, and Java. Harlow: Pearson.