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

Game's objective is to pass several levels and in the last one to fight the final boss. In addition, in each level on the path of the hero, there will be various enemies who will try to stop him.

The map is basically bottom of the ocean, while there are islands which need to be avoided. In addition, the game will be horizontal, unlike Space Shooters.

Controllers are WASD for maneuvering, spacebar for shooting and "z,x,c" keys for skills.

There will be three levels. Each level has small bosses while the third level will have the final boss. The submarine will gain experience while killing the enemies in order to level up and get better weapons.

Similar Games:

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

2.1. Gameplay

The main hero which is a submarine has health and energy. If health goes to zero, the submarine is destroyed and the game ends. In addition, there will be different enemies with different skills and attacks. At the same time, the submarine will also have different skills which have a cooldown and require some energy. The energy can be filled collecting the shiny objects on the map.

2.2. Levelling

Leveling of the submarine and leveling of the map will be independent. Submarine will level up by killing enemies and reach next level of the map by killing the final boss that appears at the end of each map. There will be three different maps in the game with three different final bosses. Each level has small bosses while the third level will have the final boss that has more health and strength compared to other bosses. The player should beat the final boss in order to win the game.

2.3. Weapons

The submarine levels up by killing the enemies and getting experience. These levels up affect submarine in a way that it improves submarine's weapons and unlock some skills, which directly affects the killing speed and the killing efficiency of the submarine. There will be 5 levels for submarine, and each leveling up will require more experience than the previous one. Each leveling up will have different effects on the weapons such as increasing the number of shooting in a millisecond, adding extra different directions to the weapons so it can shoot the different directions at once, increasing the number of weapons etc.

2.4. Skills

The skills of the submarine will be affected as same as weapons. Thus, the skills will be locked at the beginning of the game, however, each leveling will unlock some skills which will help submarine to kill more enemies at once, or to survive etc. In addition, 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 at each leveling up, which will automatically help the user to play the game more easily.

2.5. Enemies

Enemies will try to prevent you from reaching the end of the level in different ways. They are classified by three according to their level of difficulty: 1) common creatures, there are different species of them in large numbers at each level; 2) bosses, they are stronger than common creatures and have several abilities, they occur at the end of each level except the last level; 3) the mega-boss is the strongest enemy that the players meet only at the end of the last level. As already mentioned, there are different types of ordinary creatures: 1) some of them do not have the ability to shoot, but they deal damage to submarine upon collision, 2) others stay away from the submarine and shoot at it.

3. Requirement Specificaiton

3.1. Functional Requirements

3.1.1. 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.

3.1.2. 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.

3.1.3. 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.

3.1.4. 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.

3.1.5. View Credits

The user will be able to see the developers of the game directly from the main menu, otherwise, credits scene will not bother the user during gameplay.

3.1.6. Change Settings

The user will be able to change music and game buttons from change settings menu which could be reached from the main menu in the beginning of game.

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

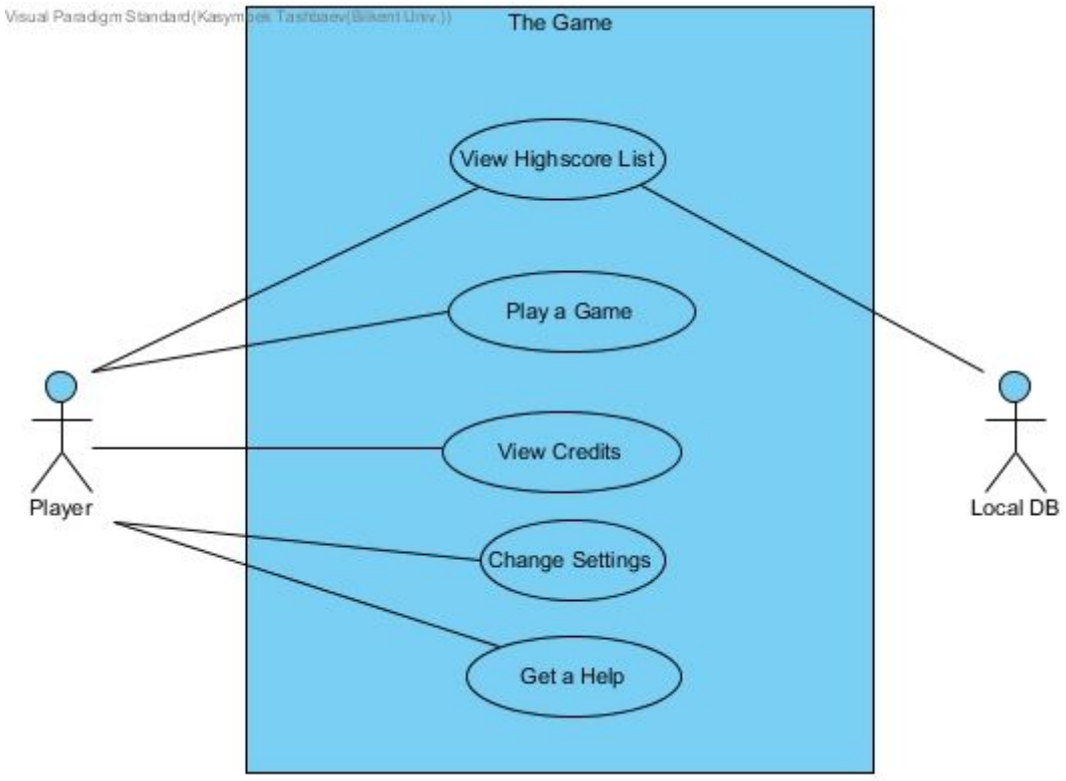
BufferStrategy object will be used to construct interface more smoothly.
Different submarine pictures will be used to make animation.

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. View High Scores

Use Case Name: View 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: -

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.

4.1.2. View Credits

Use Case Name: View 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: -

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.

4.1.3. Play the Game

Use Case Name: Play Game

Primary Actor: Player

Stakeholders and Interests:
-Player aims to complete all levels and make the highest score.
-System keeps the score of the Player.

Pre-condition: For first running, game settings are set as default. If Player changes game settings, adjusted settings will be saved and used by System.
Post-condition: If Player gets a score which is high enough to be in high score list, high score list will be updated by System.

Entry Condition: Player selects "Play 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 from the first level.
3.Player plays the level until he reaches the end and defeat the boss
4.System grants access to next level.
5.Player starts playing next level.
Player repeats the steps 3 – 5 until all levels are completed or Player is destroyed.
6.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.
7.System returns to Main Menu.

Player repeats the steps 1 – 7 if he wants to play game again.

Alternative Flows:

3A. Player tries to reach the end and defeat the boss to complete the level:

3A.1. Player starts to move by using the keyboard.

3A.2. Enemy appears and tries to damage.

3A.3. Player releases projectile forward by shooting or using the available skills to damage the enemies.

3A.4. If the projectile hits the enemy, enemy's health is reduced by the power of the weapon of the Submarine. If its hitpoint is lower or equal to 0, it is destroyed.

3A.5. Submarine 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 destroyed enemy type.

3A.7. Player prevents being destroyed by evading enemies and their projectiles, and restoring his health.

Player follows steps 3A.1 – 3A.7 to reach the end of level.

3A.8. The boss, or the final boss, if it the last level, appears and tries to destroy the Player.

3A.9. Player kills the boss and completes the level.

3B. Player collects the power ups during game:

3B.1. Power up appears.

3B.2. Players moves towards to the power up.

3B.3. Player tries to collect power up by colliding with it.

3B.4. System removes the power-up, whether user collects it or power up disappears.

3B.5. If user collects power up the System makes necessary changes to apply the feature of power up.

- *Whenever a power-up occurs during the game, steps 3B1-3B5 are applied.*

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

A.1. Player presses the proper key from keyboard to pause the game.

A.2. System pauses the game.

A.3. System shows the pause menu.

A.3.1. If Player selects resume game from the pause menu, System resumes the game.

A.3.2. If Player selects “Return to Main Menu” from the pause menu, System directs Player to Main Menu.

A.3.3.If Player selects “Change Game Settings” Change Game Settings use case is applied.

A.3.4. If Player selects “View Help”, View Help use case is applied.

A.3.5. If Player selects “Return to Main Menu”, System displays Main Menu.

A.3.6. If Player selects “Exit Game”, System closes the game and returns to desktop.

4.1.4. 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, changing the keys to play.

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

Pre-condition: Initially, game settings will be set as default. If Player changes game settings, changed settings will be saved and used by the system.

Post-condition: Game settings are changed.

Entry Condition: Player selects “Change Settings” button from the main 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. If 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. If 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. If (s)he selects yes, the settings are saved, not otherwise.

B.3. Player returns Main Menu.

4.1.5. View Help

Use Case Name: View Help

Primary Actor: Player

Stakeholders and Interests:

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

Pre-conditions: Player should be in Main Menu and click “View Help” option.

Post-condition: -

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

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

Success Scenario:

1.Player selects “View Help” from the main menu.

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

Alternative Flows:

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

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

B.2. Player returns to the main menu.

4.2. Dynamic Models

4.2.1. Sequence Diagrams

4.2.1.1. 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

- Submarine
- Small enemy
- Big enemy
- Boss Manager
- Enemy
- Game Engine
- Collusion Manager
- Object Random Location Manager
- Main Menu
- Map
- Weapons
- Health
- Skills
- Bullet
- Health Regeneration
- Cooldown Manager

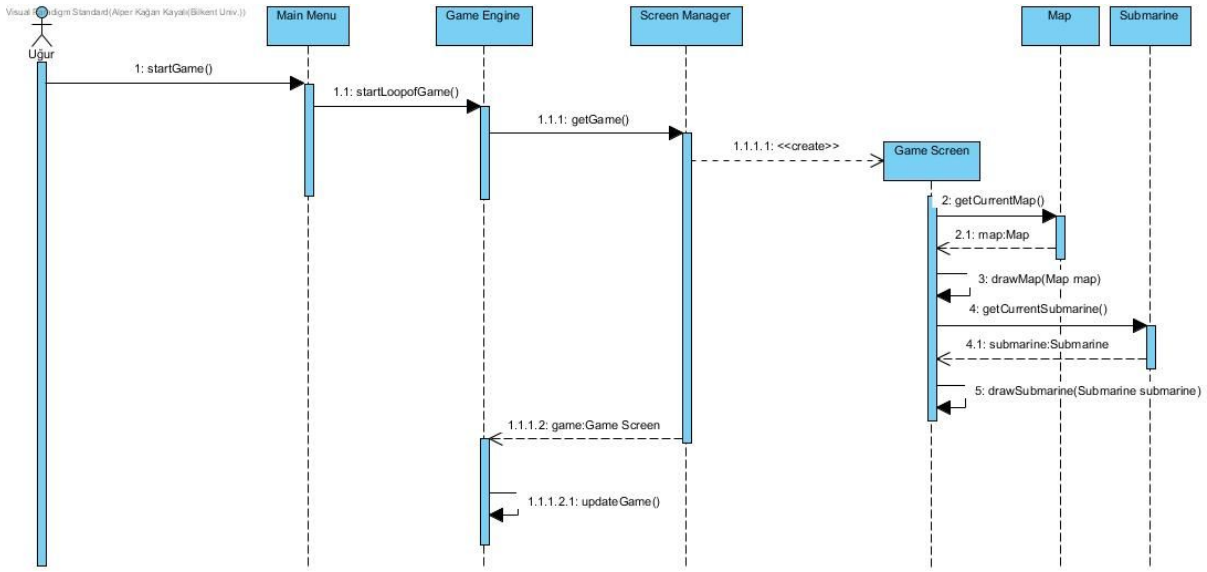


Figure 4.2.1.1 shows the sequence diagram which explains start game scenario

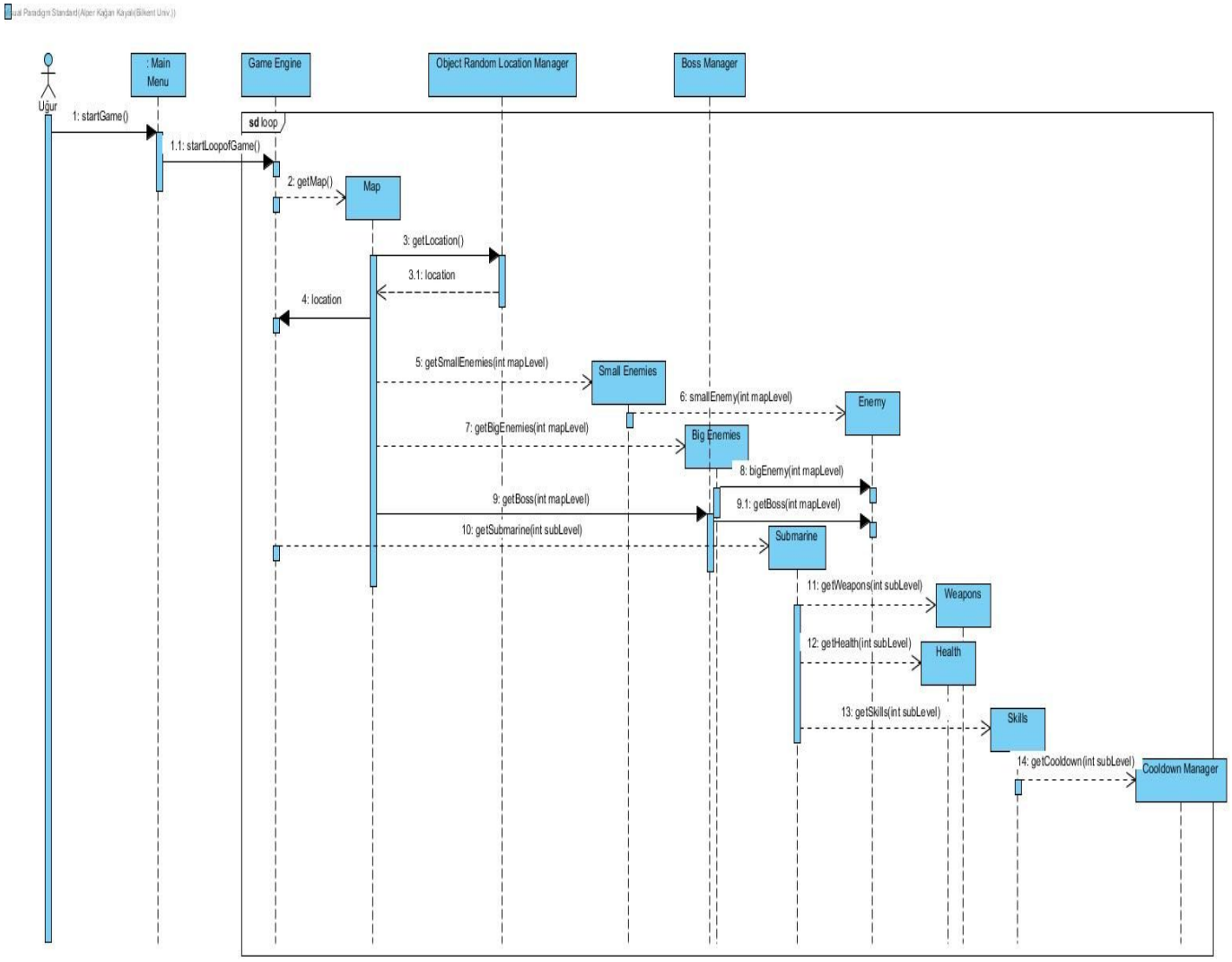
Here MainMenu is a boundary object by which user can perform some activities like starting the game, viewing credits, changing settings. In addition, ScreenManager handles creating full-screen window and graphics issues. The locations of the game objects are stored in GameMap class, GameMapManager is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

4.2.1.2. 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.



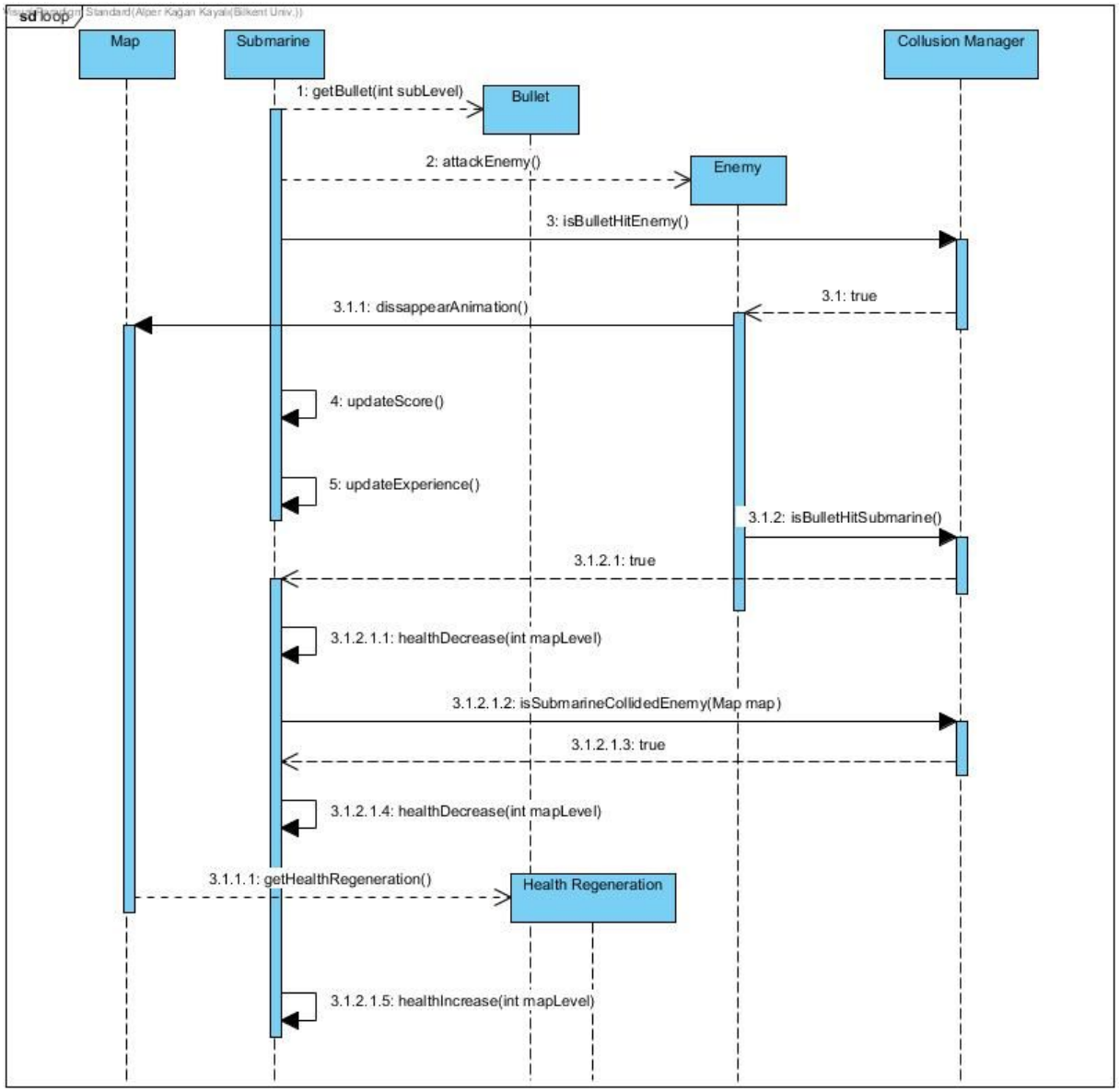


Figure-4.2.1.2 shows the sequence diagram which explains Gameplay

4.2.1.3. Change Settings

Following sequence diagram illustrates the scenario explained below:

Scenario 1: Player Uğur presses the Change Settings button from Main Menu. The System displays current settings and buttons to change the settings. Player changes music volume, music, and first action key. Then, player presses back to the menu button, System applies new settings by 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.

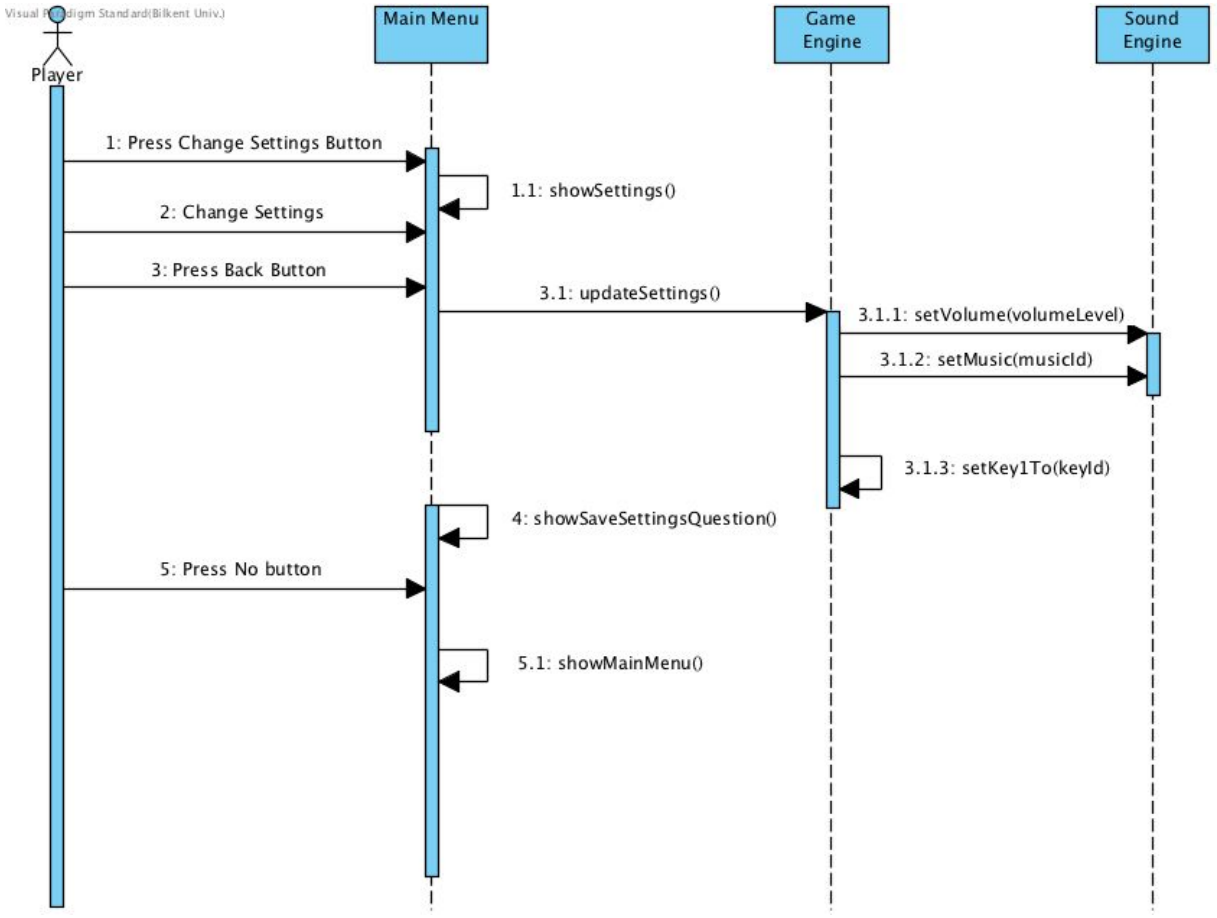


Figure-4.2.1.2 illustrates the change settings sequence diagram

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.

Scenario 2: After new settings are applied. If Uğur pushes yes button when the system asks if he wants to save settings, Game Engine saves current settings.

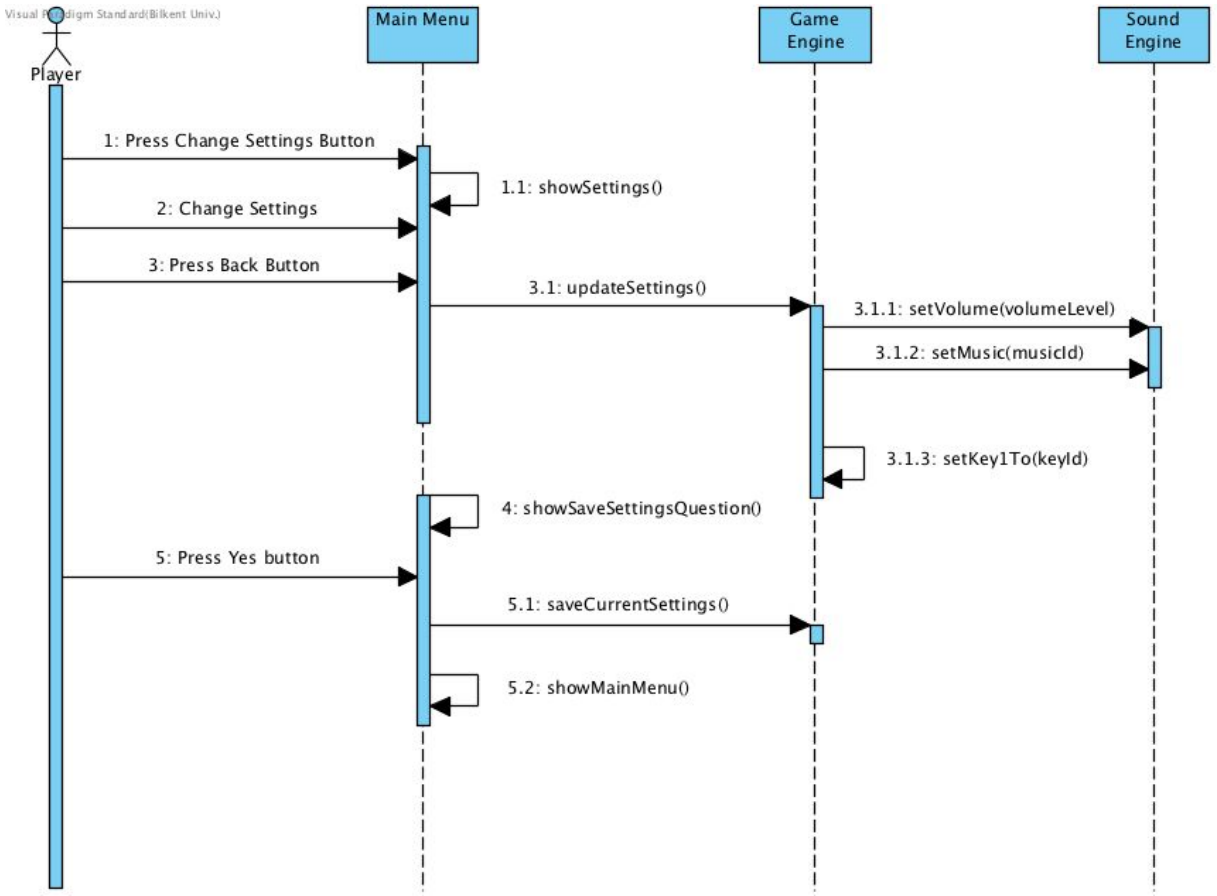


Figure-4.2.1.3 (Alternative -1) illustrates save settings scenario

Scenario 3: If Uğur has changed the game settings before and wants to change settings to default, he presses default button after pressing change settings from the menu. Game Engine sets default settings as current settings. Then displays the main menu.

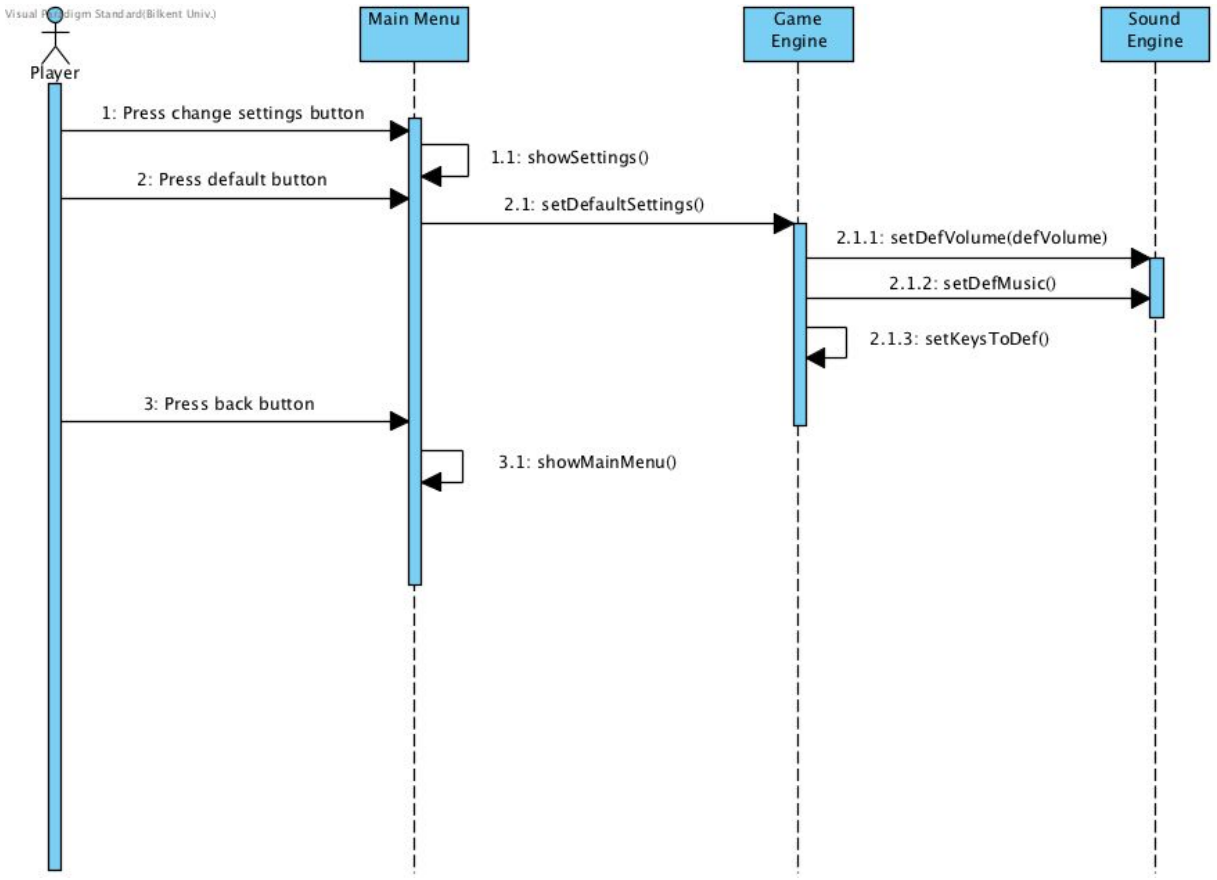


Figure-4.2.1.4 (Alternative-2) illustrates setting settings to default settings scenario

4.2.2. Activity Diagram

This diagram shows how system maintains gameplay.

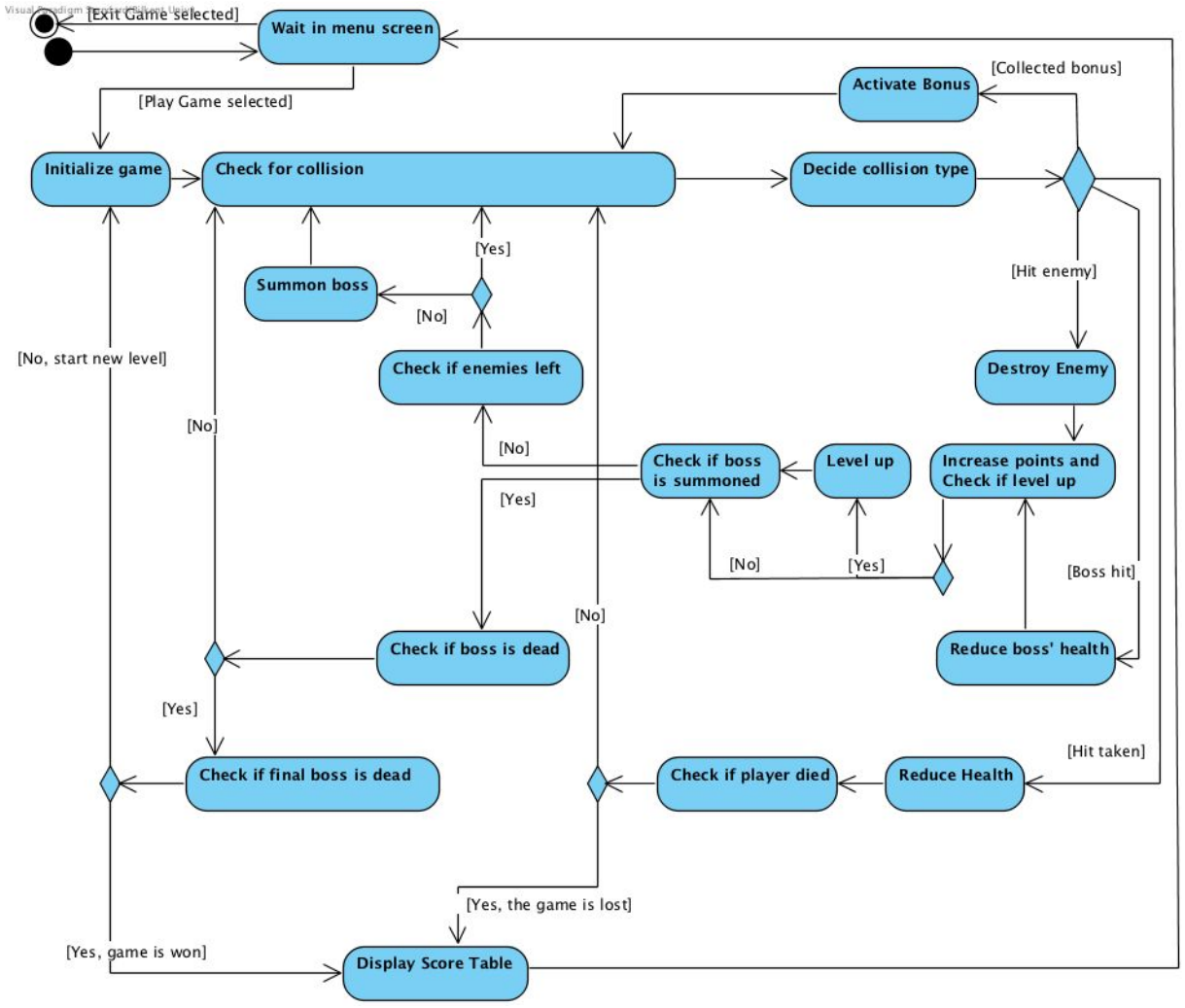


Figure 4.2.2 Illustrates Activity Diagram of in-game activities

When the user selects Play 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, player starts to attack to the enemies and system checks if there are any collisions or not. If the player hits enemies, points section will be updated. When player collects a certain amount of points, 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.

4.3. Object and Class Model

Note: All classes have getters and setter for all atributes

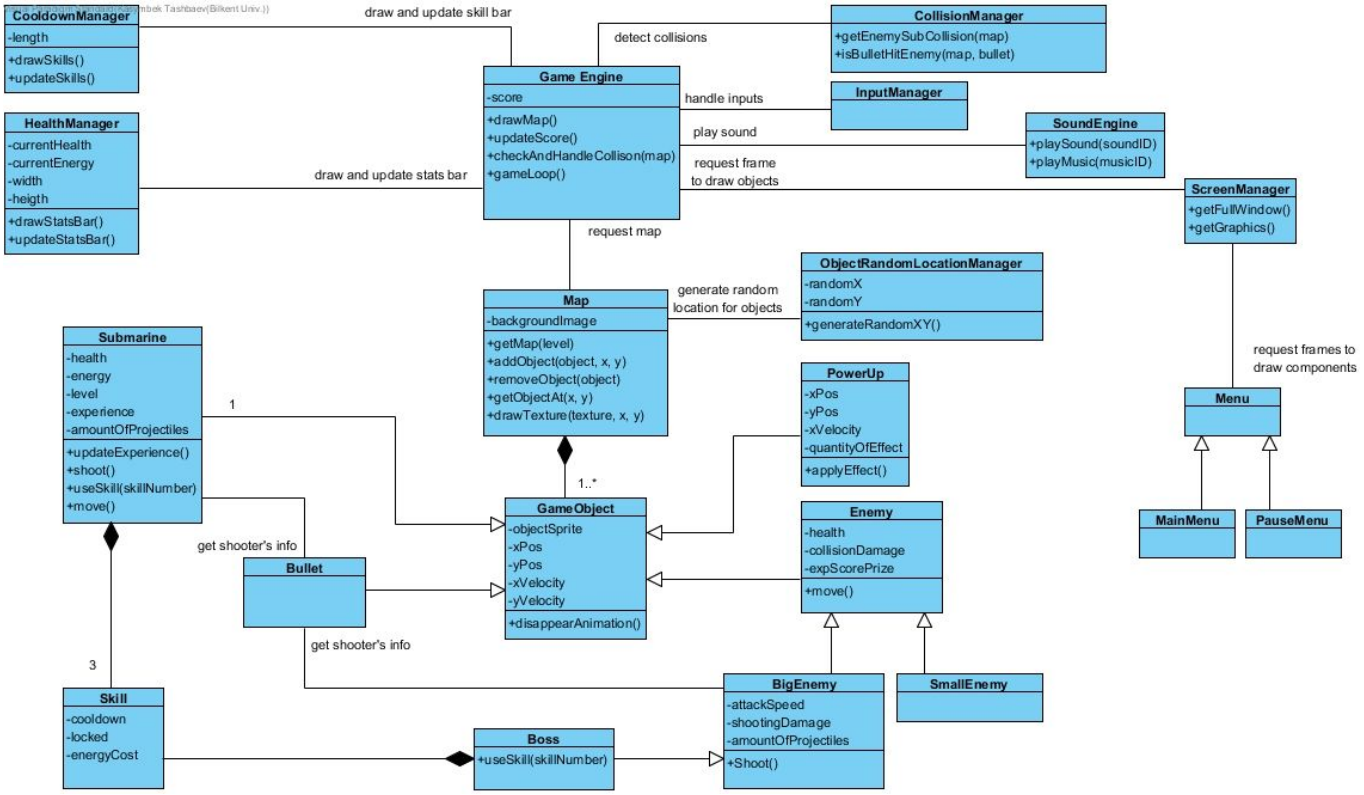


Figure-4.3 Class diagram of Sea Adventures game

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

Brief explations about classes are given in below section.

- **GameEngine** class that manages the game loop and relations between other objects.
- **CollisionManager** manages if collision occurs between submarine, enemies or bosses and projectiles. Or if player hits enemies.
- **Enemy**(Small and Big enemies) objects of enemies in game. Will carry information about their power, health and location.
- **Submarine** will carry information about submarine's health, location and current level.
- **Boss Manager**(Small and Final bosses) Same with enemy objects.
- **Skills** holds the skills that are available and sends the submarine the skills it can use.
- **Weapons** holds the weapons that are available and sends the submarine the weapons it can use
- **Map** holds the current info of the map
- **Health** health of the submarine and enemy
- **Cooldown Manager** holds the cooldown time needed for each skill at current time.
- **Health Regeneration** health regeneration objects in the game.
- **Object Random Location Manager** sends random location to the map in order to create an enemy.

5. User Interface

5.1. Navigational Path

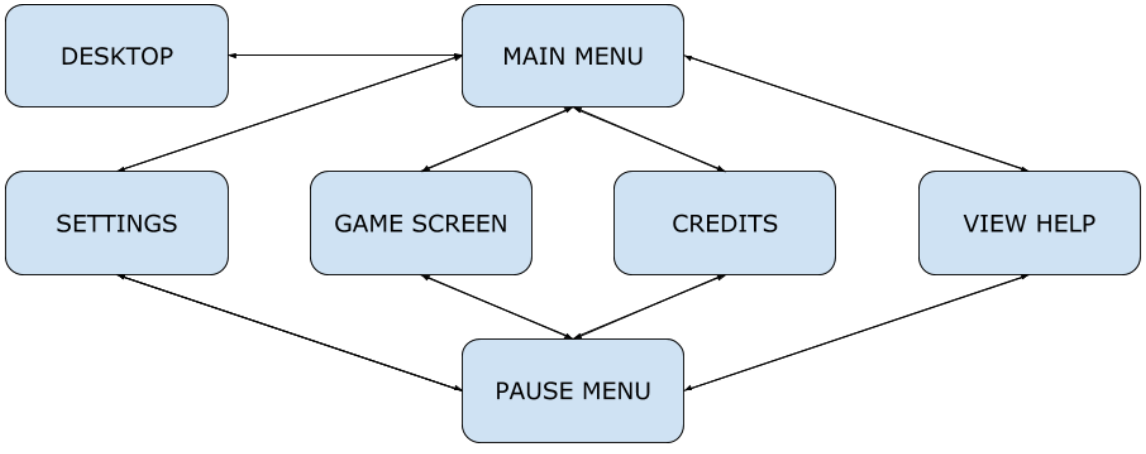


Figure 5.1 Illustrates the Navigational Path of Sea Adventures

5.2. Screen Mock-ups

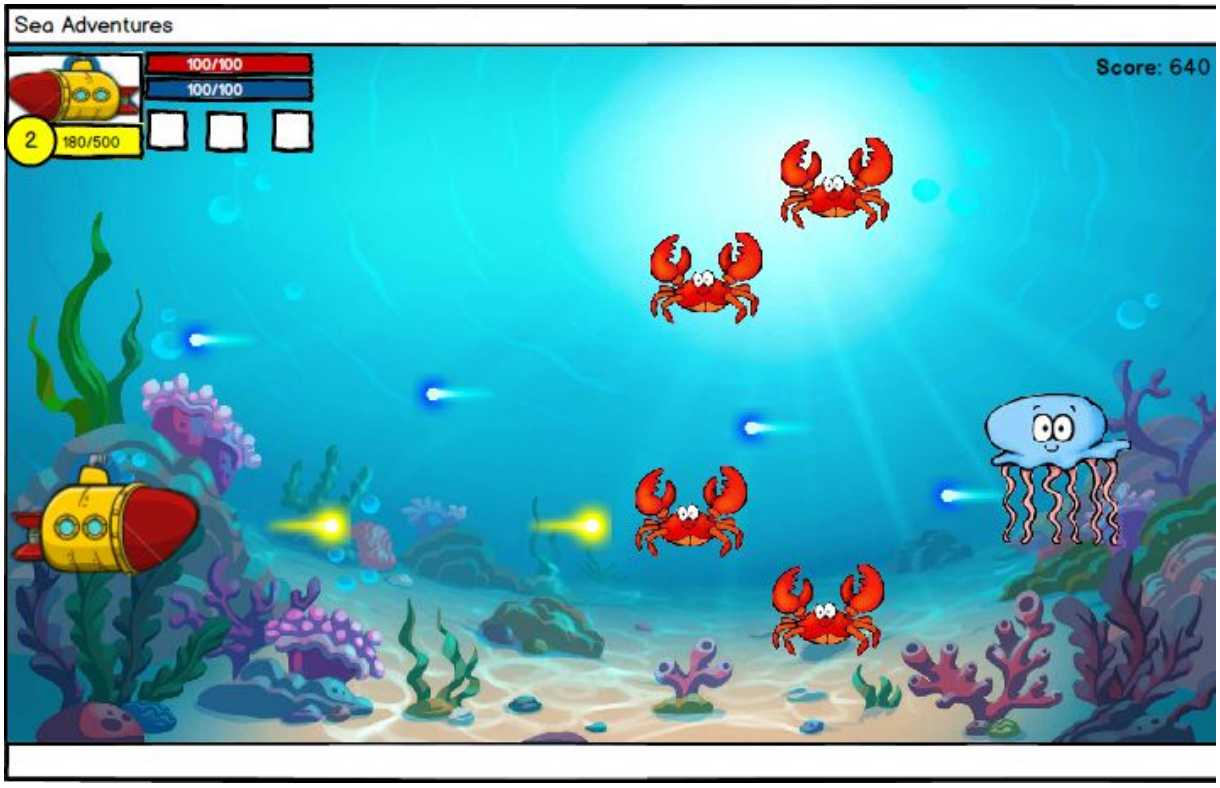
5.2.1. Main Menu

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

- Play Game
- View Highscores
- View Help
- Change Settings
- Credits
- Exit Game

Figure 5.2.1 Screen shot of Main Menu

-Play Game: When player selects “Play Game” option from main menu, game starts with current settings from the current level player is in.



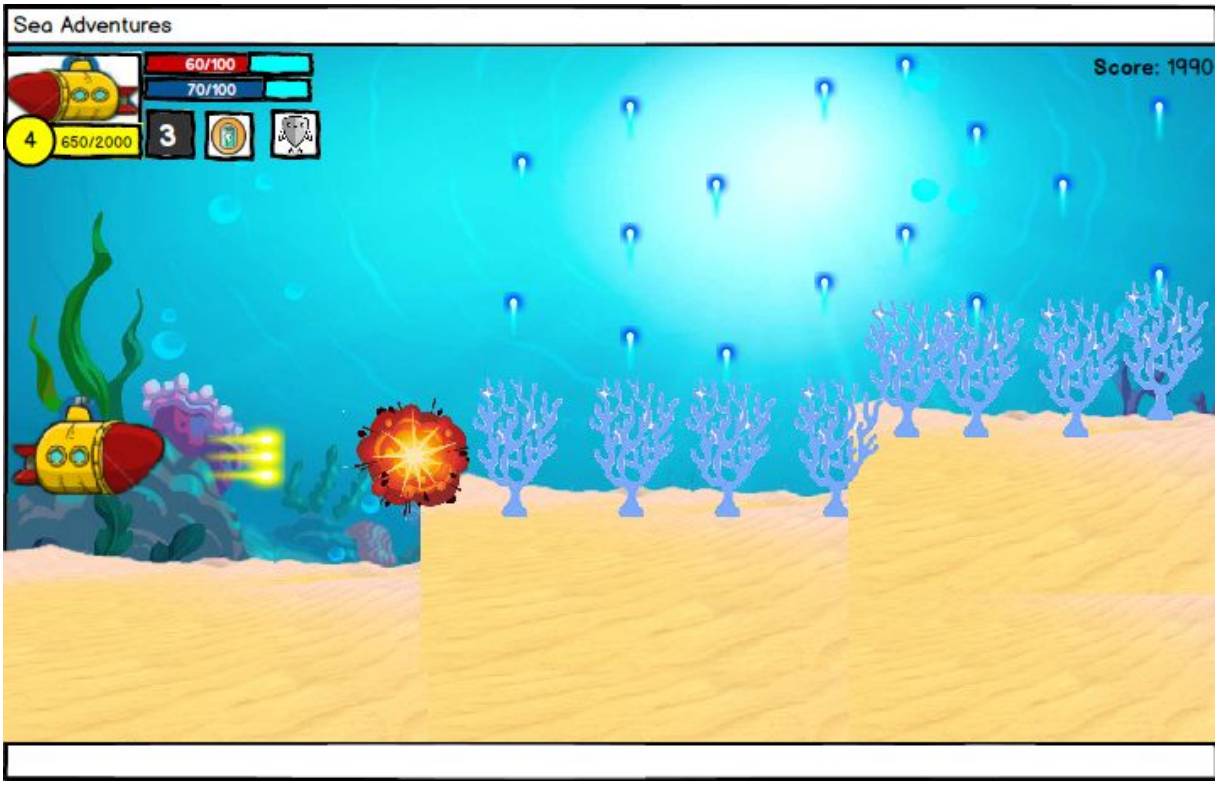


Figure 5.2.1.1 Screen shots of gameplay

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

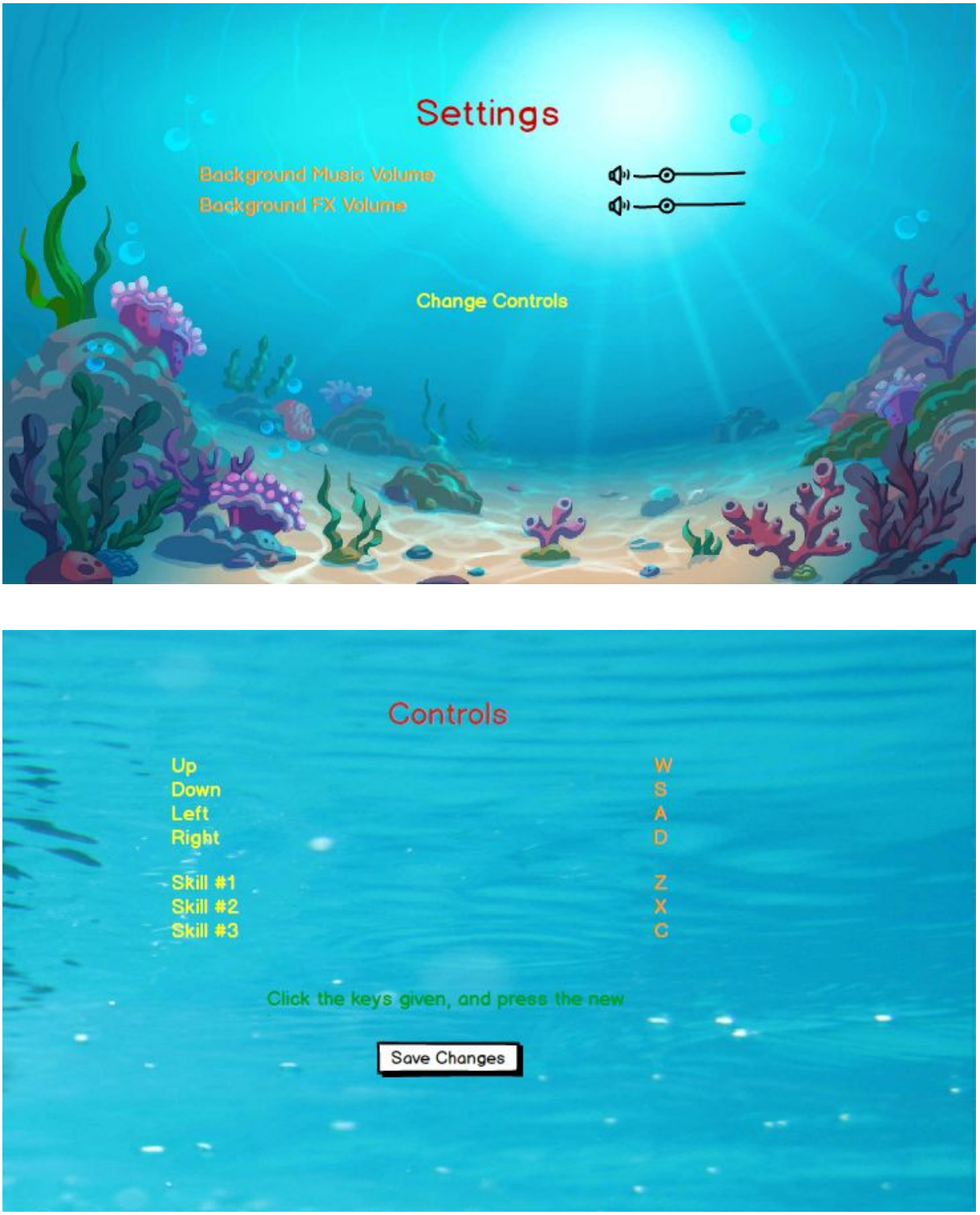


Figure 5.2.1.2 Screen shot of Change Settings

-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.

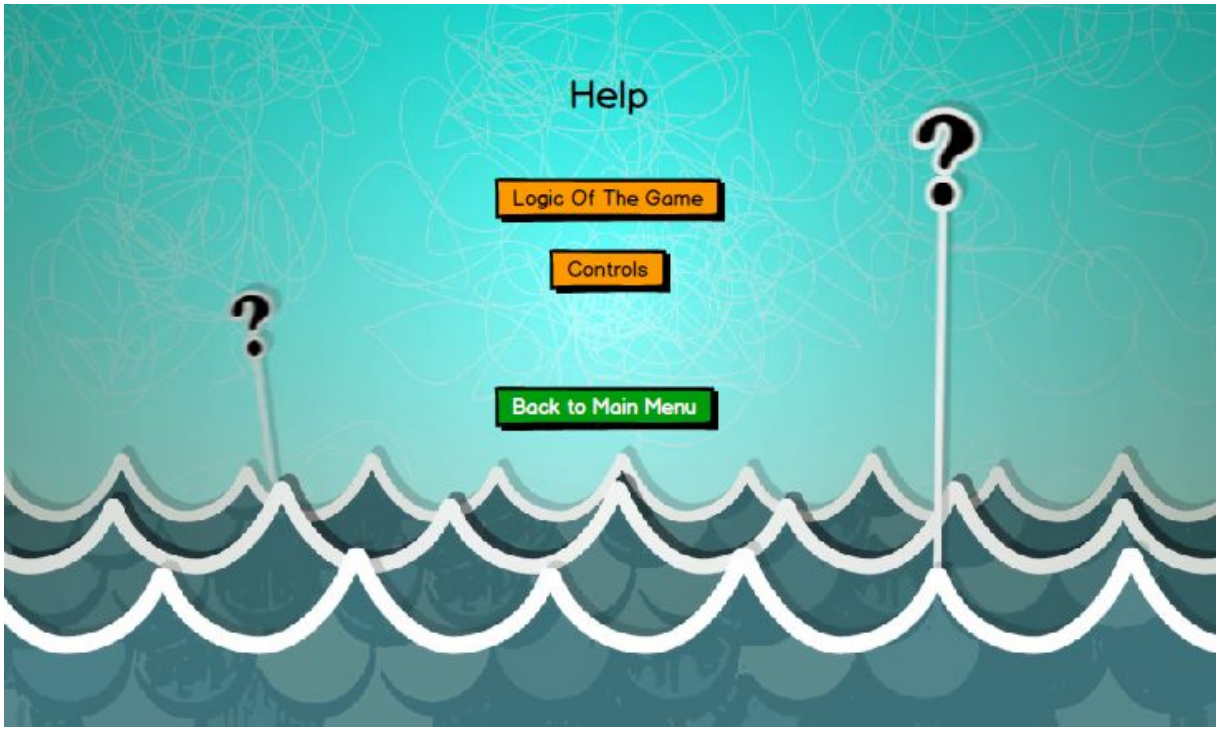


Figure 5.2.1.4 Screen shot of Help

-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.

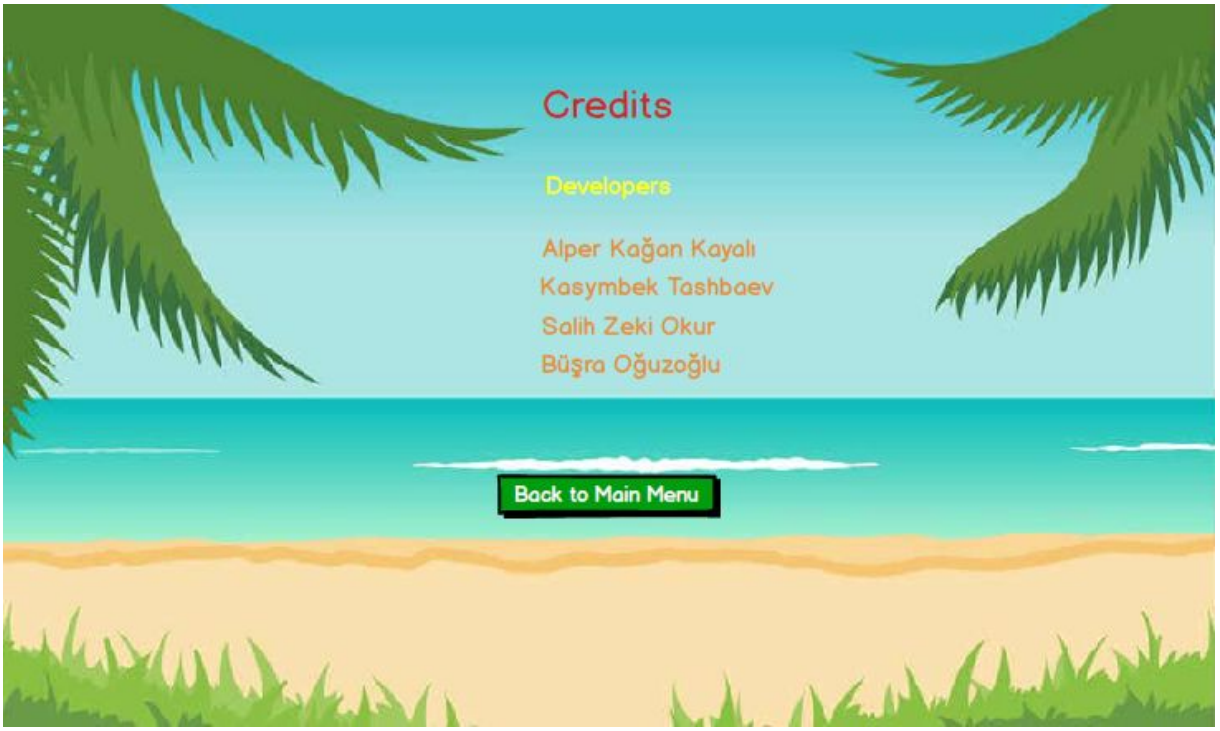


Figure 5.2.1.5 Screen shot of Credits

-Exit Game: When player wishes to exit 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 is 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 game, he should select “Back to Main Menu” then he can exit game. (Figure 5.2.2)

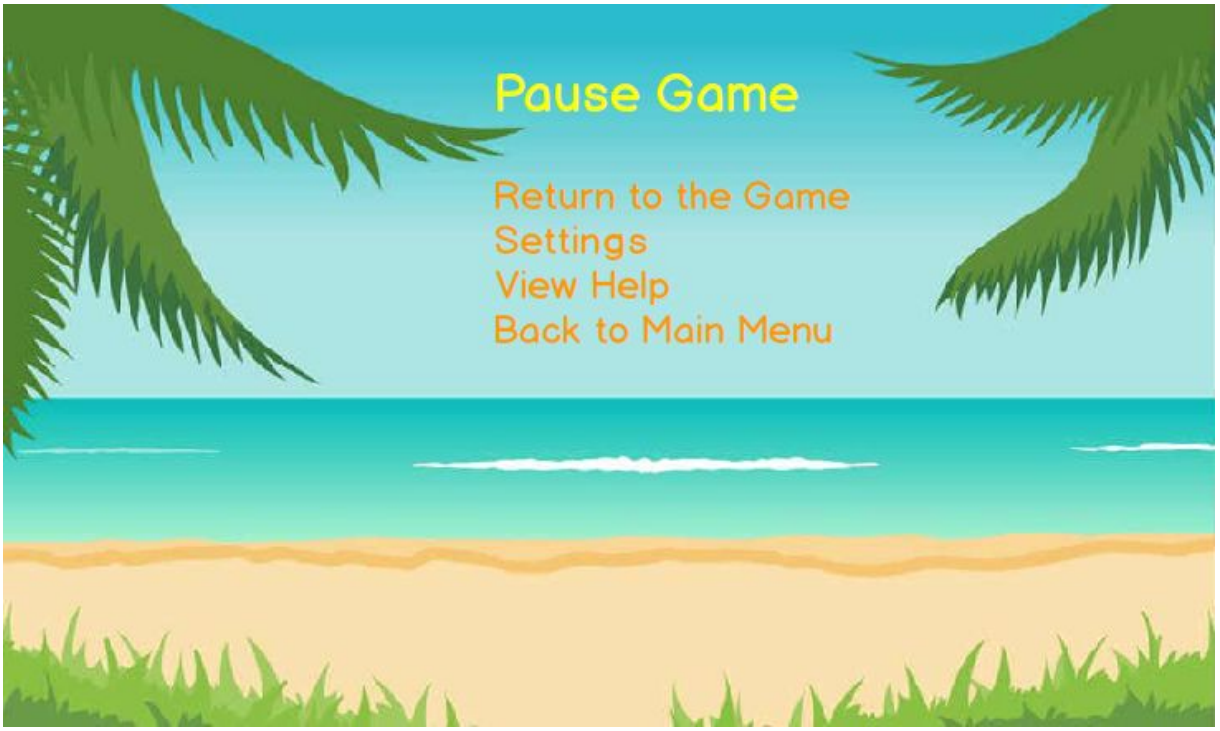
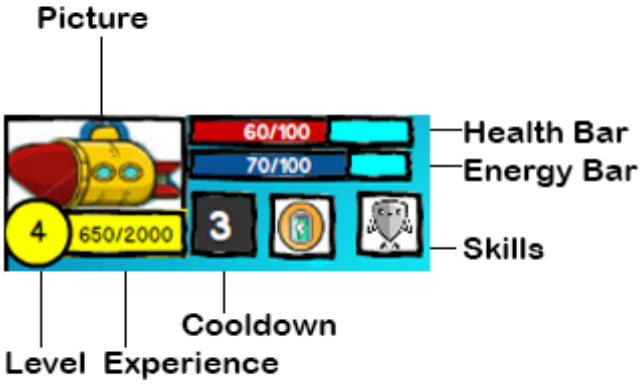


Figure 5.2.2 Screen shot of Pause Menu

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)

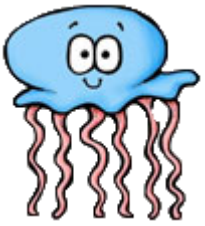
5.2.5. Enemies

Different kinds of enemies:

Dark Reef Coral: immobile, shoots only upward

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

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



Pipsqueak jellyfish: stays far from the submarine, and releases projectiles towards the submarine

Underwater Mine: slowly moves in the direction of the submarine and at a certain distance from the submarine explodes and causes damage in a large radius

Cowardly octopus: stays far from the submarine, and at each shoot releases 8 projectiles towards the submarine

5.2.6. Power ups:

Repair kit: regenerates health

Fuel: restores energy.

5.2.7. Bosses

End map bosses for first two level: (will be added later)

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 funcitonal 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 activity diagram, we tried to clarify how our game system works in possible scenarios. While doing the analysis of 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 be interacted. 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 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 player to see the developers of the game, fourth option is view help, in order to help 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 have a back to the main menu option, which will allow user to go to the main menu. Finally, user can pause the game which will help 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.

Navigational path shows scenes that can be reached by user in game. Which include game scene, settings, view help, credits scenes and main menu, pause menu scenes which can

be reach to and reached by previous four scenes. User enters to and exits from game through main menu screen.

8. References

Submarine's sprite taken from:
https://www.gamedeveloperstudio.com/graphics/viewgraphic.php?item=1r4390441v055y2r6u

Other sprites are open-source, taken from Google Image