BİLKENT UNIVERSITY

Department Of Computer Engineering

CS 319- Object Oriented Software Engineering

FIGHT OR FLIGHT

Analysis Report

Section 1

Group 1D

- MİHRİ NUR CEREN
- ADAHAN YALÇINKAYA
- BERK ERZİN
- EMRE SÜLÜN

TABLE OF CONTENTS

Açıklamalı [1]: bu tablo başlıkları değiştirdikçe otomatik güncelleniyor. değişiklik yapılacaksa bu tabloda değil başlıkların kendisinde yapılmalı

1. Introduction

1.1. Purpose of the system

The purpose is creating an entertaining and competitive game which enable users to play in both singleplayer and multiplayer mode.

1.2. Scope of the system

The system works in desktop computer on which Java is installed. Multiplayer functions supports 2 players on the same computer. So, any network connection is not required.

1.3. Objectives and success criteria of the project

Objective of this process is to create a working game that combines two different genres and provides an entertaining experience all the while learning to work on larger projects as group, utilizing what we learned through the course.

Our success criterias are,

- Successful implementation of “fighting” stages.
- Successful implementation of “running” stages.
- Successful implementation of multiplayer and singleplayer
- Successful implementation of different weapons and attacks.
- Successful implementation of different enemy kinds.
- Modular and easy to understand source code.

1.4. References

- Bruegge, B, & Dutoit, A 2014, Object-Oriented Software Engineering : Using UML, Patterns, And Java, n.p.: Boston : Prentice Hall, 2014., Bilkent University Library Catalog (BULC), EBSCOhost, viewed 16 February 2018.

2. Overview

Fight or Flight is a hybrid game which consist of both runner game elements and 2-D shooter/rpg game elements. The main aim is to survive as long as you can while getting the most amount of points, like most runner games our game ends with the player eventually dying as well.

In most runner games, the player runs while the speed of the character and the amount of points the player gains increases, the speed increasing to a level so high that the player eventually dies thus the game ends where it is the most exciting. Our game has got these runner game elements described above with one more additional feature. Instead of the game ending with the player dying at some point while running, in “Fight or Flight” the player gets to stop running and take the game to the “Fight” stage ending the “Flight” stage. In this stage you stand where you are and confront waves of orcs, undead and bosses with each wave. After the end of each wave, it is up to the player to transition back into “flight” stages or keep fighting waves to get more points. Defeating consecutive waves gains player a larger points multiplier but with each wave player also has to face tougher and more numerous enemy groups.

Overall, the point of this game to survive as long as possibly can while accumulating points. You can try to survive on your own or cooperate with a friend since the game can be played together with up to 2 characters.

2.1 Gameplay

Açıklamalı [2]: Duplicate yazılar düzeltilecek

Singleplayer: The game begins with the “Flight Stage” with player character starting his run through an obstacle course like level. In this stage player can only control their character’s vertical movement by pressing “W” key for up and “S” key for down. Throughout the level, there will be obstacles to hamper the player. By pressing “W” or “S” the player will change the characters vertical position to avoid the obstacles or grab items. The player can press “TAB” to switch between stages.

In the “Fight” stage the player gains full control over his character’s movements. “W” to go up, “A” to go left, “S” to go down and “D” to go right on the screen. Adding to that, in this stage player will have to fight against waves of enemies. “Left Mouse Button” is used for standard and the “Right Mouse Button” is used for special attacks. The player is able to pause the game anytime by pressing “ESC”.

Multiplayer: Multiplayer mode allows another player to take control of a second character. Player 2 uses the arrow keys to move his character instead of “W”, “A”, “S”, “D”. Melee attack is designated to “O” and ranged attack is “P” for Player 2. Switching between stages and pausing the game are the same key for both players.

2.2 Levels

The game has 2 levels which the player decides to switch between until the player dies.

Flight level: The Game starts with this level. Random obstacles appear as the player automatically runs through an ever ending map. Armor and ammo will spawn along the way randomly. The rewards will not spawn entirely random, the faster the character gets, spawn chance of rewards increase. Player only has one health in this stage. Failing to avoid an obstacle would result in death. This level ends either by death or by the player switching the level.

Fight level: This level always appears after a Flight level. The automatic running of the player comes to an end with the player now gaining full control over the characters movement. There will be waves of enemies from wave 1 to wave n, n being the number where the player decides to switch the level or die. Player will initially encounter small number of basic enemies. Number of enemies and the difficulty to beat these enemies will increase with each wave with special kind of enemies spawning in further waves. Armor and ammo will randomly spawn in this level as well with spawn chance increasing with each wave. This level ends either by death or by the player switching the level.

2.3 Armor, Power-ups, Enemies, Skills and Characters

- **Paladin Armor**
  - **Judgement Armor:** Basic Paladin armor. Increases health by 50. Holy Light deals 10 additional damage.
  - **Lightsworn Armor:** Strong Paladin armor that comes equipped with *Ashbringer*. Increases health by 75. Melee deals 25 additional damage. Holy Light deals 25 additional damage. Can only be worn by Arathras the Pure.
  - **Ashbringer:** A legendary paladin sword. Increases melee damage by 25. Cannot be wielded without Paladin armor. Cannot be wielded by Death Knights.

- **Death Knight Armor**
  - **Betrayer of the Light!:** Harnessing dark energy, the character “Darion” becomes fully equipped with Death knight armor. Holy Light is replaced with Death Coil. Melee deals 50 additional damage.

- Scourgeborne Armor: Strong death knight armor. Melee deals 25 additional damage. Death Coil deals 25 additional damage. Cannot be picked up before “Betrayer of the Light!”.

- **Power ups:**
  - Lesser Healing Potion: Increases health by 25.
  - Ancient Health Potion: Increases health by 50.
  - Valorous Health Potion: Increases health by 100.
  - Lesser Mana Potion: Increases health by 25.
  - Ancient Mana Potion: Increases health by 50.
  - Valorous Mana Potion: Increases health by 100.
  - Altar of the Lightbringer: Gives the Paladin Max Health and Mana.
  - Altar of Darkness: Gives the Death Knight Max Health and Mana.

- **Enemies:**
  - Undead: Basic enemy. Low health and damage. Can only deal melee damage.
  - Orc Warrior: Skilled fighter. Moderate health and damage. Can only deal melee damage.
  - Abomination: Slow moving monstrosity with very high damage and health. Can only deal melee damage.
  - Gargoyle: A flying creature with low damage and health. Can deal ranged damage.
  -Boss: An elite enemy with extremely high damage and health. Has different skills than regular mobs.

- **Paladin Skills:**
  - Melee: The paladin deals damage with his main weapon.
  - Holy Light: A ranged attack that sends a bolt of holy energy to the target. Effective against undead enemies.
  - Divine Shield: The paladin becomes invulnerable for 5 seconds.

- **Death Knight Skills:**
  - Melee: The death knight deals damage with his main weapon.
  - Death Coil: A ranged attack that sends a blast of unholy energy to the target. Effective against orcs.

- Bone Shield: The death knight becomes immune to all damage for 5 seconds.

- **Characters:**

  -Arathras: An experienced paladin of the Silver Hand, mentor of Darion. Stats of the character are below(without any armor or power-ups):

  - Health: 100
  - Mana: 100
  - Melee Damage: 10
  - Ranged Damage: 10

  -Darion: A young paladin of the Silver Hand, student of Arathras. Initially a paladin, can later become a death knight. Stats of the character are below(without any armor or power-ups):

  - Health: 100
  - Mana: 100
  - Melee Damage: 10
  - Ranged Damage: 10

3. Functional requirements

- Players should be able to view highscores.
- Players should be able play this game with either multiplayer or singleplayer.
- Players should be able to switch between “fight” and “run” levels.
- Players should be able to use weapons and special powers.
- Difficulty level should increase over time with “fight” stage containing more enemies and the run stage becoming faster.
- The game should have different and interesting enemies and bosses.
- Players should be able to change the background music of the game.
- Players should be able to change control buttons on keyboard.
- Players should be able to see a manual containing information about the game on the “help” screen
- The game should have non-repetitive and interesting fighting mechanics.
- Players should be able to pause the game to either continue playing, changing settings, viewing help, or exiting the game.

4. Nonfunctional requirements

4.1. Usability

- The system should have a simple interface so that any user who knows basic English can play easily.
- The system should show keyboard controls and allow users to change them.
- The game does not need installment.

4.2. Reliability

- The software needs to be bug-free.
- Possible bugs should be foreseen and prevented.
- The game should not crash since progress is an important aspect of this game.
- The user’s data should be kept if a crash were to occur.

4.3. Performance

- The system should provide at least 30 fps for a satisfying experience.
- The system should support 2 concurrent users.
- The response time of all of the movements and attacks should be short.
- Since the fight stage has infinite waves with each wave containing more enemies, the game should be able to handle large quantities of objects.
- Since the background changes very rapidly in the “run” stage, it should change smoothly without any delays.

4.4. Supportability

- The system should be able to maintain itself if new updates and new game mechanics were to be implemented.

4.5. Implementation

- Users should have a monitor that supports minimum 800x600 resolution.

4.6. Interface

- Users should interact with the system via a keyboard and a mouse
- The interface should be user-friendly, visually satisfactory and intelligible

4.7. Packaging

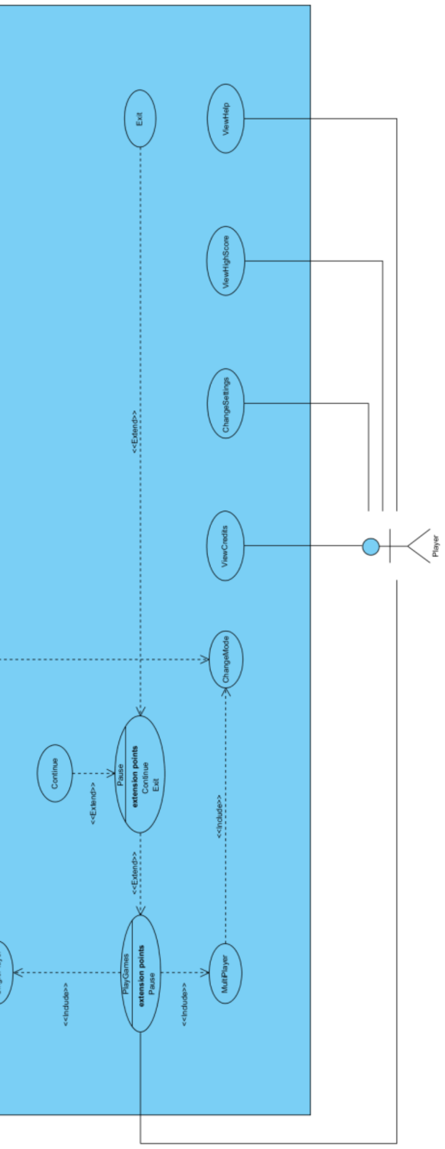
- Game should be runned via a jar file.
- No installation is required because running the jar file is sufficient.

4.8. Legal

- The system is open source and hosted on GitHub
- The system is licensed under the Apache License 2.0.

5. System models

5.1. Use case model



Exit
ViewHelp
ViewHighScore
ChangeSettings
ViewCredits
ChangeMode
Player
<<Extend>>
Continue
<<Extend>>
Pause
extension points
Continue
Exit
<<Include>>
<<Extend>>
PlayGames
extension points
Pause
MultiPlayer
<<Include>>
<<Include>>

Figure 5.1.1 Use Case Diagram Of The System

5.1.1 **PlayGame**

Use case name: PlayGame

Participating actor: Player

Entry condition: Player is on the main menu.

Exit condition:

- Player ran out of your lives, OR
- Player has won the game, OR
- Player can choose exit by pausing the game.

Main flow of events:

1. Player starts the game.
2. Player chooses singleplayer or multiplayer.
3. Player starts from the flight level.
4. Player’s score is displayed on the screen.
5. Player pauses the game.
6. Player returns to the main menu.

Alternative flow of events:

1. Player loses all his/her health and the game is over, player returns to the main menu.

5.1.2 **ChangeMode**

Use case name: ChangeMode

Participating actor: Player

Entry condition: After pressing play button.

Exit condition: Player returns to the game.

Main flow of events:

1. Player is playing the flight level.
2. Player clicks on the “Change Mode” button to switch the level.
3. Player starts to play the other level (fight level).
4. Player clears a wave of enemies.
5. Player clicks on the “Change Mode” button to switch the level back to the flight level.
6. Player dies in the flight level. The game is over.

5.1.3 **SinglePlayer**

Use case name: SinglePlayer

Participating actor: Player

Entry condition: After pressing “play game” button.

Exit condition: Player starts to play the game.

Main flow of events:

1. Player clicks on the “single” button.
2. Player starts to play the game.

5.1.4 **MultiPlayer**

Use case name: MultiPlayer

Participating actor: Player

Entry condition: After pressing “play game” button.

Exit condition: Players start to play the game.

Main flow of events:

1. Player clicks on the “multiplayer” button.
2. Players start to play the game.

5.1.5 **ViewHighScore**

Use case name: ViewHighScore

Participating actor: Player

Entry condition: Player is on the menu.

Exit condition: Player returns to the menu.

Main flow of events:

1. Player clicks on the “View High Score” button.
2. High scores are displayed.
3. Player returns to the main menu.

5.1.6 **ViewHelp**

Use case name: ViewHelp

Participating actor: Player

Entry condition: Player is on the menu.

Exit condition: Player returns to the menu.

Main flow of events:

1. Player clicks on the “View Help” button.
2. Help is showed.
3. Player returns to the main menu.

5.1.7 **View Credits**

Use case name: ViewCredits

Participating actor: Player

Entry condition: Player is on the menu.

Exit condition: Player returns to the menu.

Main flow of events:

1. Player clicks on the “View Credits” button.
2. Contributions are displayed.
3. Player returns to the main menu.

5.1.8 **ChangeSettings**

Use case name: ChangeSettings

Participating actor: Player

Entry condition: Player is on the menu.

Exit condition: Player returns to the menu.

Main flow of events:

1. Player clicks on the “Change Settings” button.
2. Player changes settings.
3. Player returns to the main menu.

5.1.9 **Pause**

Use case name: Pause

Participating actor: Player

Entry condition: Player is in the game.

Exit condition: Player returns to the game.

Main flow of events:

1. Player clicks on the “Pause” button.
2. Pause options are displayed.
3. Player returns to the game.

Alternative flow of events:

- Player can choose to start a new game.
- Player can exit the game.

5.1.10 **Continue**

Use case name: Continue

Participating actor: Player

Entry condition: After pressing pause button.

Exit condition: Player returns to the game.

Main flow of events:

1. Player clicks on the “Continue” button.
2. Player returns to the game.

5.1.11 **Exit**

User case name: Exit

Participation actor: Player

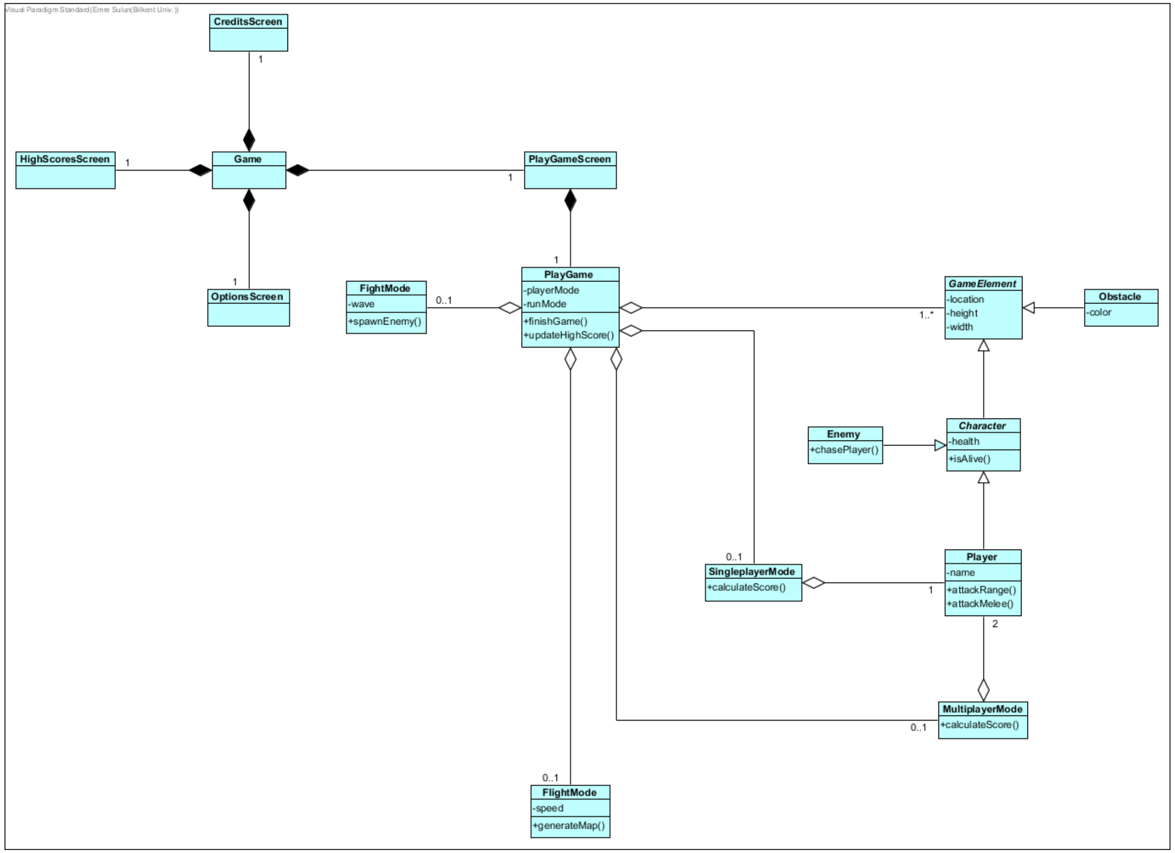
Entry condition: After pressing the pause button.

Exit condition: Player exits the game.

Main flow of events:

1. Player clicks on the “Exit” button.
2. Player exits the game.

5.2. Object model


Visual Paradigm Standard(Emre Sulun(Bilkent Univ.))
CreditsScreen
1
HighScoresScreen
1
Game
PlayGameScreen
1
OptionsScreen
1
1
PlayGame
-playerMode
-runMode
+finishGame()
+updateHighScore()
FightMode
-wave
+spawnEnemy()
0..1
GameElement
-location
-height
-width
1..*
Obstacle
-color
Character
-health
+isAlive()
Enemy
+chasePlayer()
Player
-name
+attackRange()
+attackMelee()
1
2
SingleplayerMode
+calculateScore()
0..1
MultiplayerMode
+calculateScore()
0..1
FlightMode
-speed
+generateMap()
0..1

5.3. Dynamic models

5.3.1. Sequence diagrams

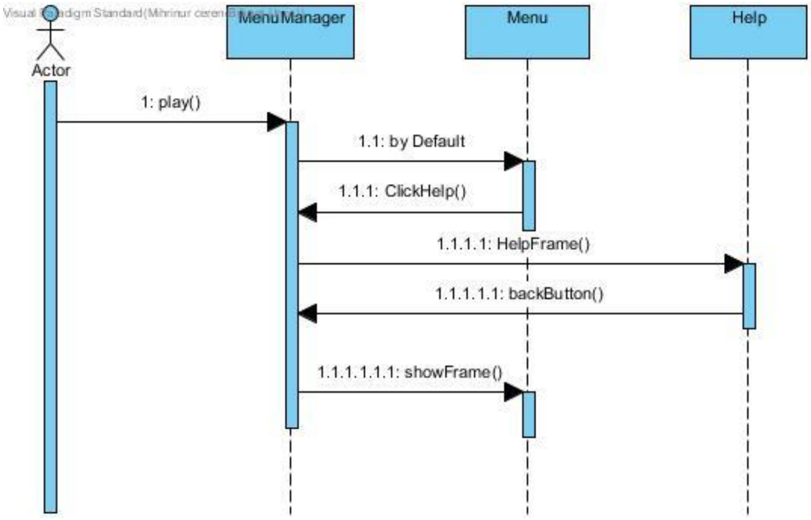


Figure 5.3.1.1: Illustrates the scenario for Help

Scenario 1# -Help

If the player wants to learn how the game is played, ”Help” button can be used. On the help frame, there are some information about the game. If the player wants to go to main menu, they can use the back button.

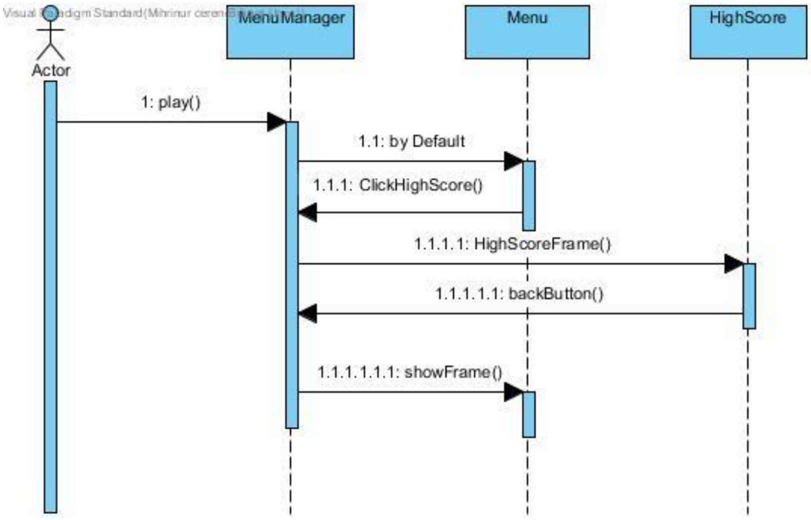


Figure 5.3.1.2:Illustrates the scenario for High Score

Scenario 2# -HighScores

If the player wants to see the high scores, the “high score” button on the main menu can be pressed. The player is also able to see the high scores table at the end of each game.

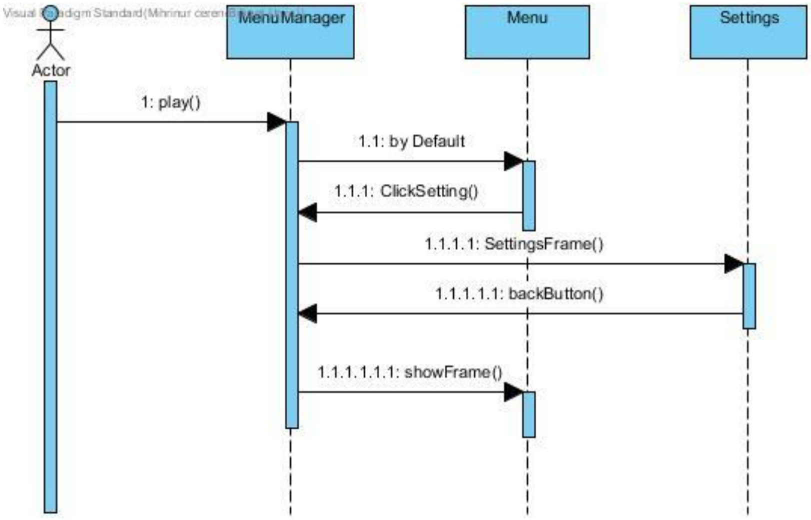


Figure 5.3.1.3:Illustrates the scenario for Settings

Scenario#3 -Settings

If players want to change settings, they can use settings button in the main menu. After changing settings, player can go back to the main menu by using back button.

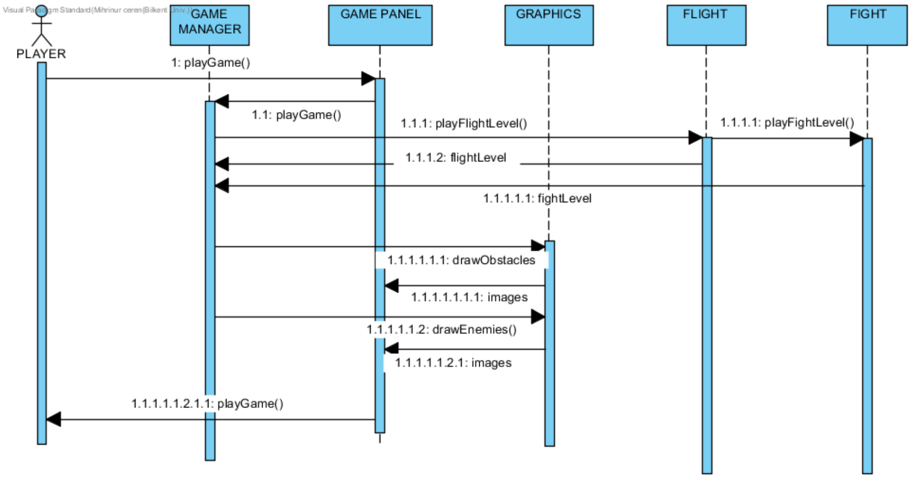


Figure 5.3.1.4:Illustrates the scenario for playing game

Scenario#4 - Play Game

The game panel is initialized after the player chooses to play the game. The game initially begins with the flight level. The player can switch the level to the “fight” level and then again switch it back to the “flight” level. Enemies or obstacles occur depending on the current level.

5.3.2 Activity Diagram

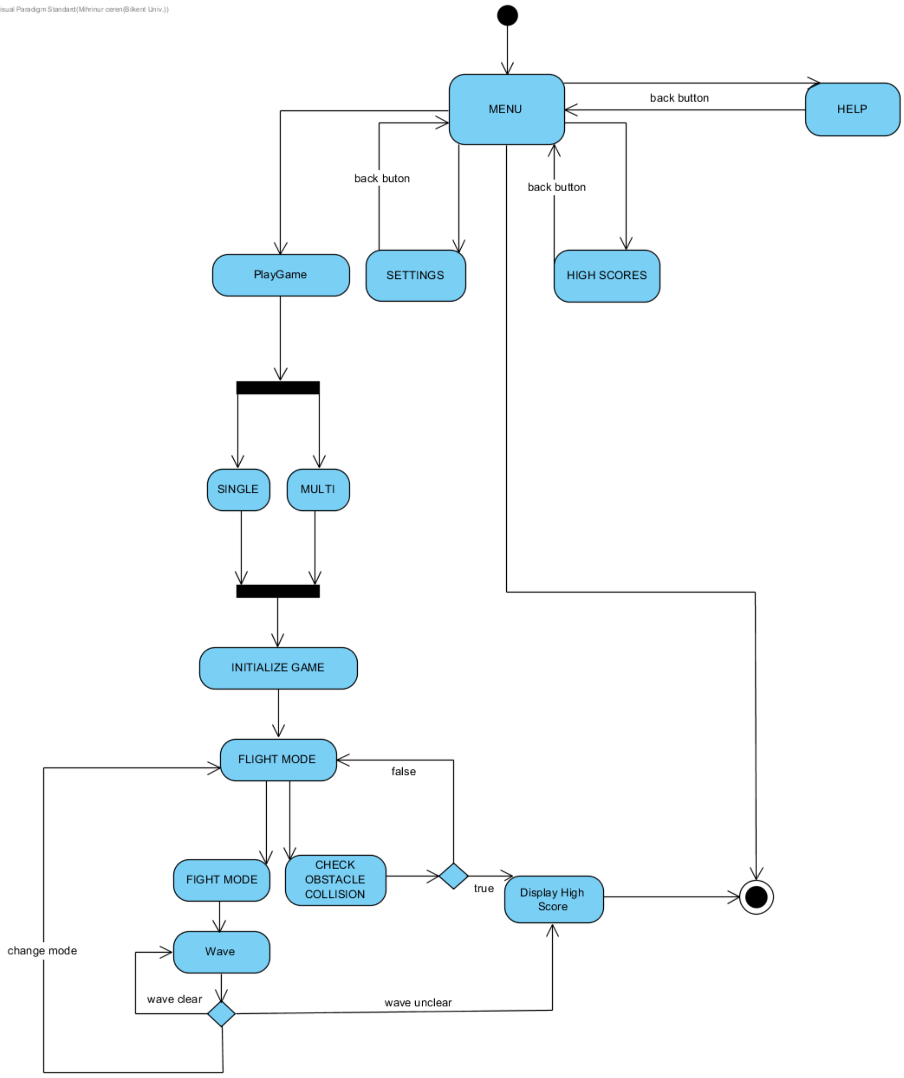


Figure 5.3.2.1:Illustrates the activity diagram for the player

5.4. User interface

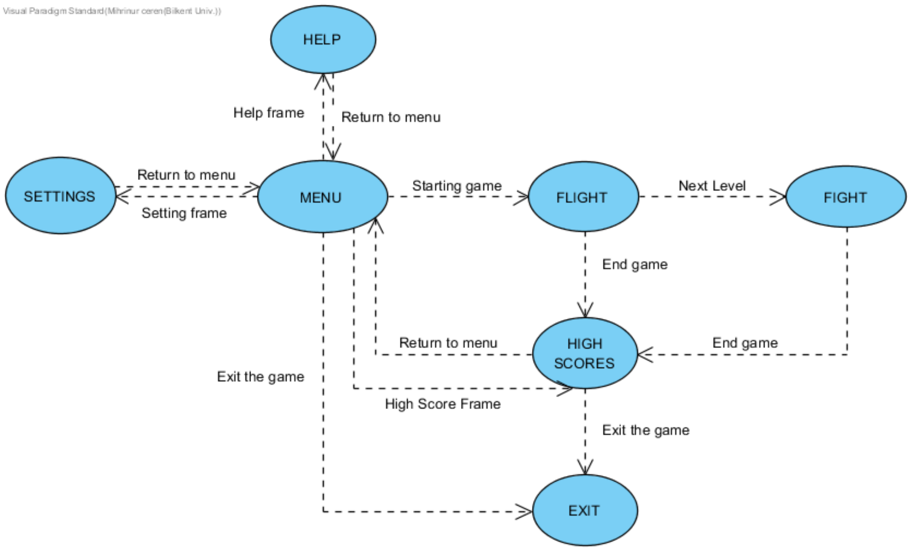


Figure 5.4.1:Illustrates the user interface

Firstly, home page is shown. The users have five options that are Help, Settings, High Scores, Exit and Play. When users press the play button, first level of game will be flight. If the users reach at the end of the flight level, next level of the game(fight) will be started. If users do not reach at the end of the flight level, they will see high scores. After showing high score, they can exit or return to menu. When they press help button, they can see some information about the game. By pressing setting button, they can be able to change what they want.If users press exit button,they can exit.

5.4.1 Mockups

Main Menu: When the game is executed, this is the first menu that the player encounters. There will be 4 buttons to press: Play, High Scores, Help, Exit.

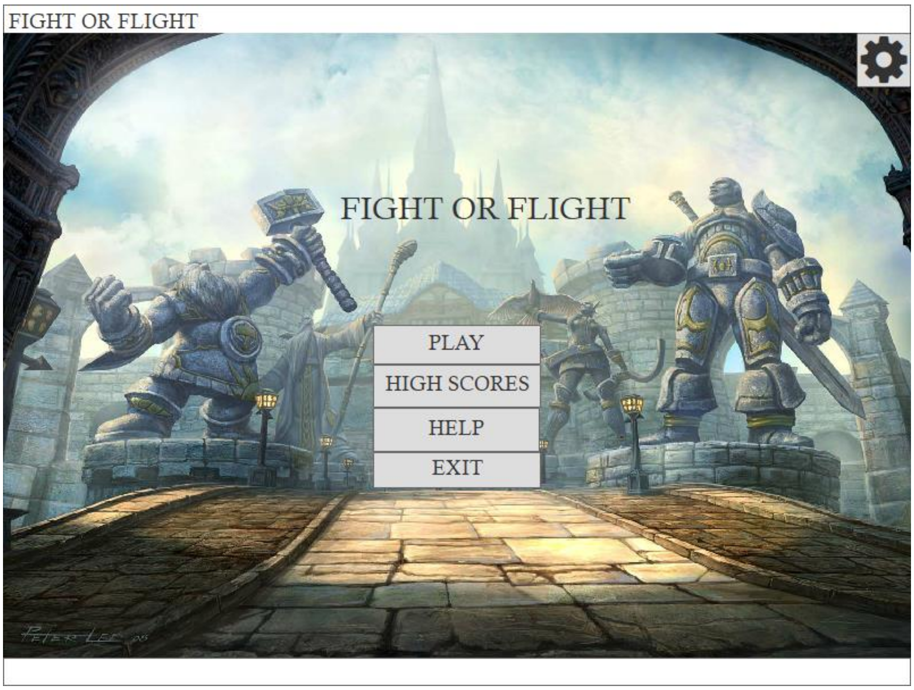


Figure 5.4.1.1: Mockup of the main menu

Single or Multiplayer Selection Screen: The player encounters this frame right after pressing the "Play" button from the main menu. By selecting "Single" or "Multi" the game is initialized with default single player settings or default multiplayer settings.

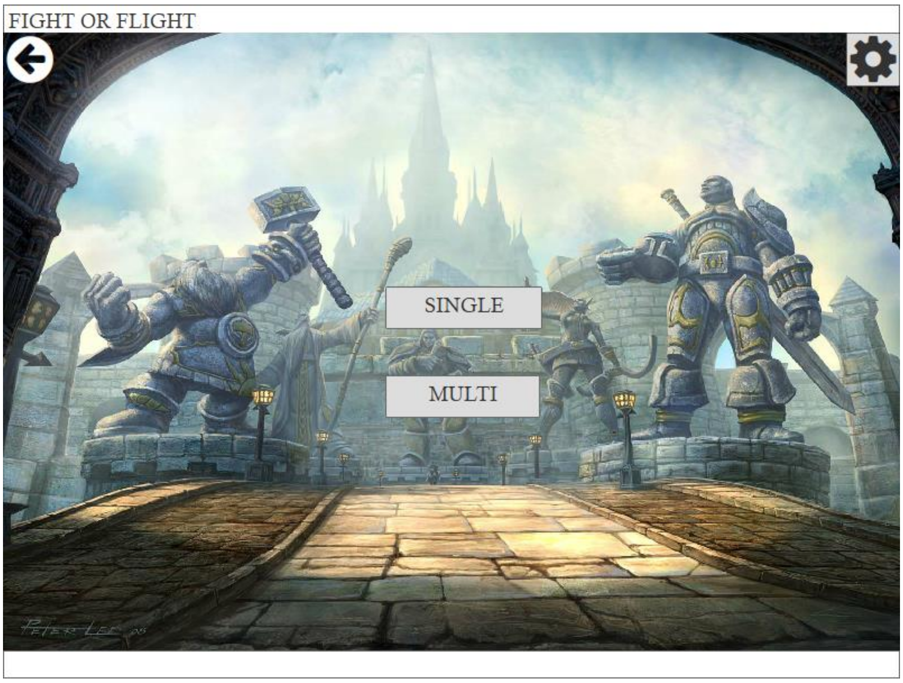
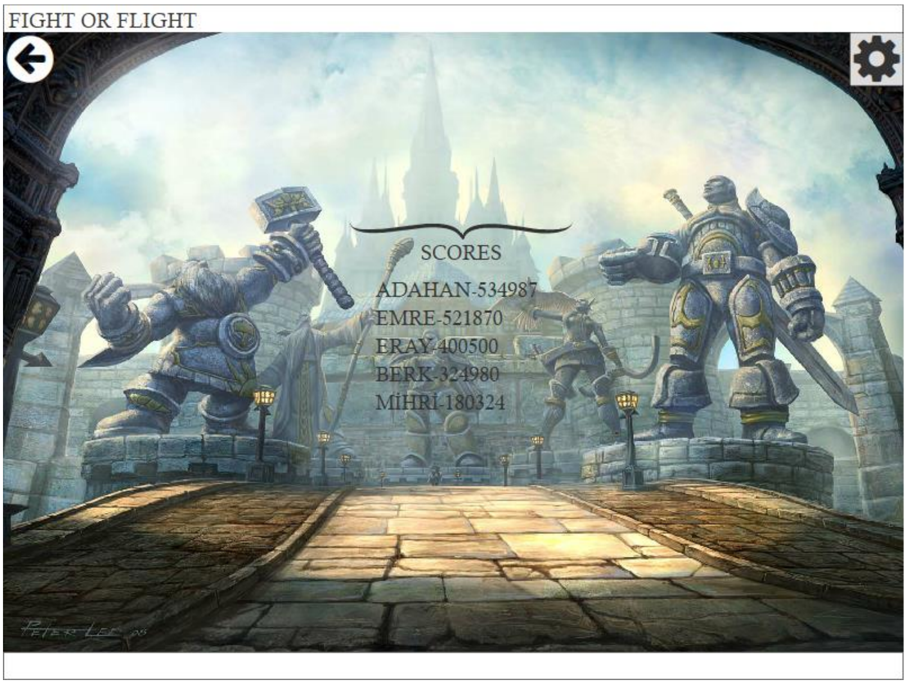


Figure 5.4.1.2: Mockup of the Single or Multiplayer Selection Screen

High Scores: This frame is displayed after the player presses the High Scores button from the main menu or after the game is over. The highest scores are displayed from top to bottom. The top 10 highest scores will be displayed. If the player's score is between this interval, the score will be displayed within the appropriate place.

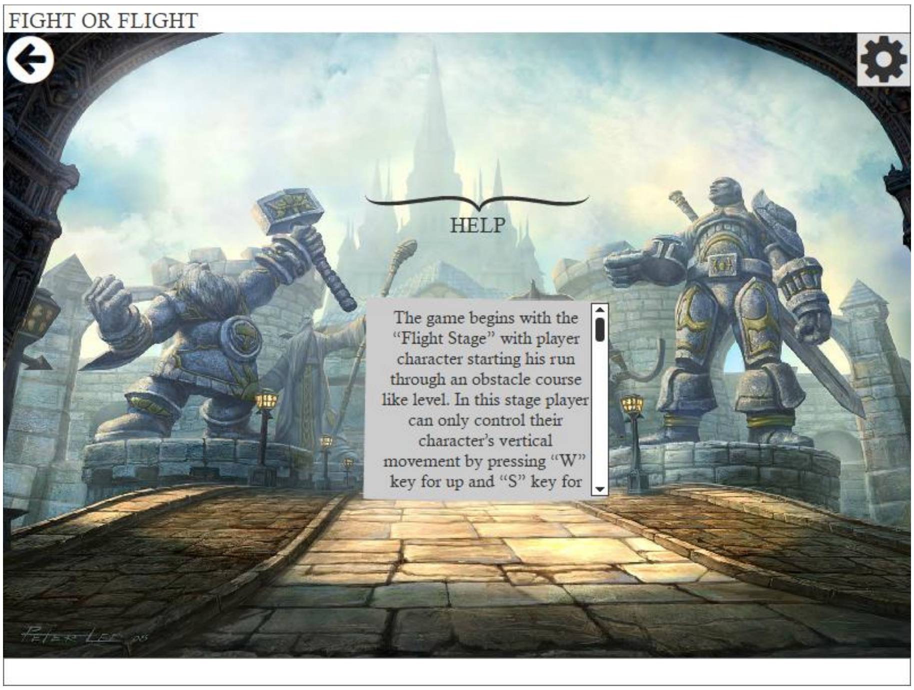


SCORES

ADAHAN-534987
EMRE-521870
ERAY-400500
BERK-324980
MİHRİ-180324

Figure 5.4.1.3: Mockup of the High Scores Screen

Help: The help screen can be reached either by pressing the help button from the main menu or the pause menu. A small manual of the game appears at the center of the screen containing all the essential information about the game. The player can scroll up and down to read this manual.



The game begins with the "Flight Stage" with player character starting his run through an obstacle course like level. In this stage player can only control their character's vertical movement by pressing "W" key for up and "S" key for

Figure 5.4.1.4: Mockup of the Help screen

Fight Stage: The mockup below represents the fight stage. The section above the game screen shows the attributes of the current level. The character's health is initially 100, when it reaches 0 the game is over. Wave denotes the current difficulty of the game. As the wave number increases, more enemies will appear. Wave also denotes the amount of points gained, if the wave number is high more points are gained. Stage denotes the current level of the game (whether it is "Fight" or "Flight") and it can be switched only when a wave is over. Score displays the current summation of the points gained throughout the game.

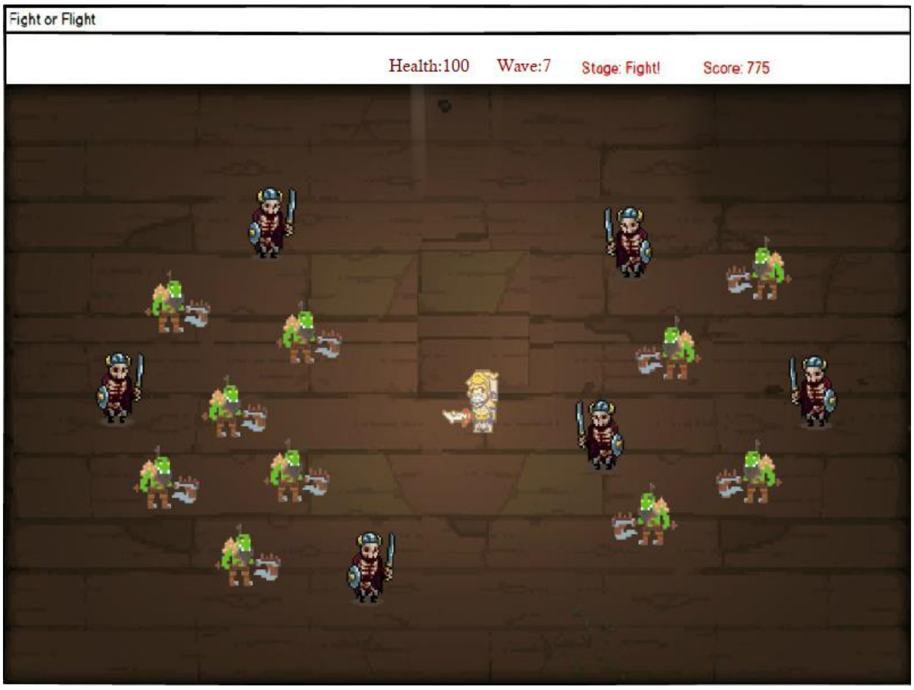


Figure 5.4.1.4:Mockup of the game screen(Fight level)

Flight Stage: Flight stage takes place inbetween fight stages. In this stage, goal is to dodge obstacles as long as possible. The longer you stay in flight stage, faster the game becomes, making obstacles harder to avoid. Yet a longer flight stage also means a larger score multiplier and might also provide bonuses for the fight stage, rewarding player for reaching higher difficulties.

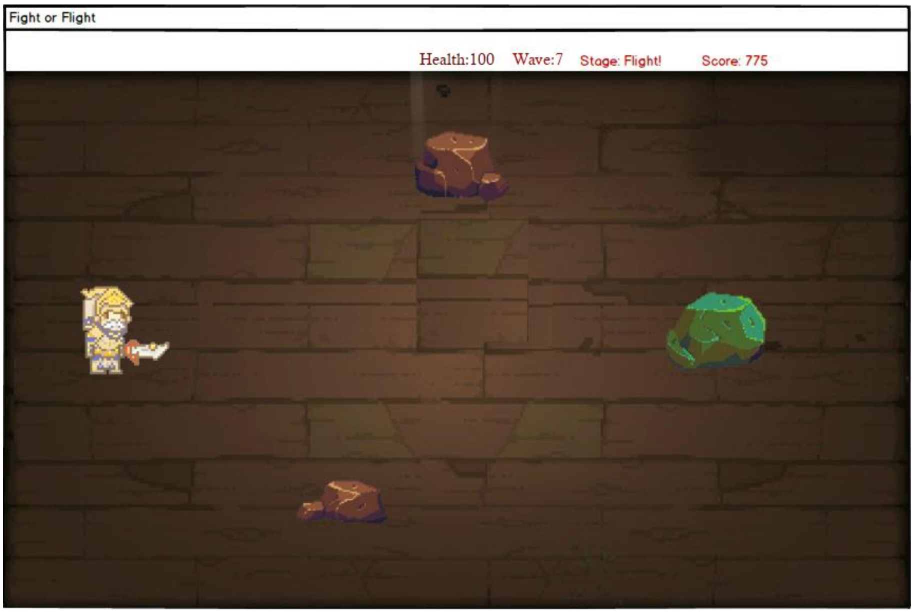


Figure 5.4.1.5:Mockup of the game screen(Flight level)

6. Glossary

- Bruegge, B, & Dutoit, A 2014, Object-Oriented Software Engineering : Using UML, Patterns, And Java, n.p.: Boston : Prentice Hall, 2014., Bilkent University Library Catalog (BULC), EBSCOhost, viewed 16 February 2018.
- https://worldofwarcraft.com/en-gb/
- http://gamelore.wikia.com/wiki/File:Stormwind_City.jpg
-