Bilkent University

Department of Computer Engineering

CS 319 - Object-Oriented Software Engineering

CS 319 Project: Catch the Deadline

Analysis Report

GROUP 2D
Argert Boja, Bikem Çamli, Hareem Larik, Umut Ak

Course Instructor: Bora Güngören

Progress/ Analysis Draft Report October 7, 2017

This report is submitted to the Moodle in partial fulfillment of the requirements of the Object Oriented Software Engineering Project, course CS319

1 Contents

Analysis Report

CS319 Project: Catch The Deadline

2 Introduction

'Catch The Deadline' is a 2D platformer arcade-style game. The main purpose in 2D platform games is usually to collect the power-ups, overcome the obstacles, completing and passing the levels successfully. "Catch The Deadline's" main purpose is similar to other games in this genre. The protagonist of the game is a student, who needs to satisfy academic expectations by overcoming a set number of obstacles. The student avoids the obstructions on his way to catching the deadline by running and jumping and eliminates distractions and enemies (home works, exams and blank sheets) by using his weapons such as "The Mighty Pen".

Throughout this report, the overview, functional requirements and nonfunctional requirements of the game will be given. In addition, the scenarios use case diagrams and UML diagrams will be shown.

3 Game Overview

"Catch The Deadline" will be a 2D platformer arcade-style desktop game, which is implemented in Java. In this game, the purpose of the player is to successfully end the levels with the highest score and within the given time. Hence the player should collect as many power-ups and prizes as he/she can and stare down the obstacles.

3.1 Features of the game

- The game will start with easy levels. As the player passes to next levels, the game will become harder and obstacles will increase. The student will grow up, therefore, will have more distractions to overcome and harder works to accomplish.

- Player will control the character using the keyboard.
- The player can pause the game both using the keyboard and pressing the pause button. - There will be time limit in the game. Hence the level should be completed within the given amount of time.

-There will be a high score table in the menu which will show the 3 highest scores in each level. The scores will be gained according to the finishing time and collected rewards.

- The high scores table will display highest scores among different players since the database will be online and this will now allow us to display scores of different players.

3.2 Power-ups

There are three different power-ups included in this game. The three of them are what each student needs: coins (money), food and sleep.

Coins: Whenever the player collects coins they will be added to his "pocket". By using this money, the player will be able to upgrade his/her weapons.

Food: Food has a great importance on well-functioning of the brain. Therefore, in order to get good grades in their home works and exams, and win the fight against their enemies, students should eat properly. In the beginning, our protagonist will start with 3 food power-ups. As the game proceeds, the student will have a lot of chances to collect more food to keep the balance of food supplies. However, in each fight against the enemies, in each hit, 1 food will be consumed. Once the student is out of food supplies, he/she will die of starvation directly, hence he/she will lose 1 life and restart the game from the beginning. If the number of lives is 0 the game will be over.

Sleep: Similarly to the food, sleep rating has an important effect on a student's performance. The student will have 3 sleep quantities in the beginning. In the same way as with the food, for each fight the student will lose one sleep quantity. Hence, if he/she doesn't collect enough "Sleep" then the player will die of sleeplessness after a few seconds unless he/she is able to get some rest(sleep power up) within the few seconds.

3.3 Enemies

Owl with homework: Home works are regular responsibilities of students in every age. In our game, our main character will have to succeed in his/her fights with the home works that are provided by an owl.

Exams: In order to finish the semester successfully, students should overcome the exams she/he will come across. The hardest enemy of the student in our game is the exam. If the character fails the fight against this enemy, he/she will lose a life and will have to start again from the beginning of the level.

Walking blank sheet: The last but not least enemy is a blank sheet. In order to overcome this enemy, the paint spray will be able to help the character to fulfill this sheet with his paint.

3.4 Weapons

Weapons will help the student overcome the enemies. Each different weapon will be used for a specific enemy.

Pen: Pens will be used to overcome the exams. By throwing the pen, students can eliminate exams and continue their Catch the deadline trip.

Eraser: The erasers will be used for home works. Just as the character of the game, all the students do mistakes. Hence the eraser is one of the best ways to deal with these mistakes. By throwing the eraser at the homework, the student will be able to eliminate it and continue his trip ahead.

Paint spray: As mentioned in the enemies section, the paint spray will be used to overcome the blank sheets.

3.5 Levels

The game will consists of 3 levels. As the levels increase, the game will become harder by increasing the number of enemies and reducing the given amount of time. These three levels will be named:

- Primary School (Easy)
- High School (Medium)

- University (Hard)

In the game overview the only change was that now we will also display the high scores of different players.

4 FUNCTIONAL REQUIREMENTS

4.1 Login

After launching the game, the player will be directed to the login window. In this window, the user can either create an account if she/he doesn't have one, or he/she can log in with his/her account. Furthermore this window will also provide the user the possibility to play as a guest, hence without creating an account.

4.2 Play Game

Catch the Deadline is a platformer-arcade-style game. The character of the game is a student who needs to catch the deadline while fighting against the enemies in order to be graduated. Student can fight these enemies by using "weapons" such as pens, erasers and paint sprays. Additionally, the student will start the game with 3 lives which he/she can use in all levels. Therefore, this game's purpose it's not only to catch the deadline in a fast way but also to collect as many power-ups and rewards as possible.

4.3 Pause

The player can pause the game. If the game is paused, the player can either chose one of the other menus such as "Settings", "Help", "Credits" or it can exit the game.

Furthermore the player can also resume to the game to where they left it.

4.4 Help

In the "Help Menu", the player can read the information on how to play the game and control the player.

4.5 Settings

In the settings window the player will be able to change the music and sound of the game by turning them on or off.

4.6 Exit

The user can exit the game in two ways. Firstly, user can use the “X” button at the right hand side of the game page. In addition, user can select the exit option from the Pause Game page.

4.7 Additional Requirements

4.7.1 Control

The user will be able to control the character of the game by using the direction keys, space bar and also the mouse for shooting. Furthermore the user will be able to choose the weapon type and shoot with “A” , “S”, “D” keyboard buttons.

4.7.2 High Scores

The player can see the highest 3 scores by pressing the High Score Table button on the home page. Scores are calculated according to the time that player finishes the game and also the power-ups that collects during a level. The highest scores will be of different players. If a player reaches a high score, his scores will be added to the database and they will be displayed in this window.

5 NON-FUNCTIONAL REQUIREMENTS

Beside the functional requirements, even the non-functional requirements are very important. A good game should also have a good performance, graphical smoothness, a good UI and also extendibility.

5.1 Game Performance

In gaming industry, game performance should be one of the main concerns. In our game, there will be a bunch of animations, like shooting, running, jumping, killing an enemy, etc. Furthermore, there will also be sound effects during the gameplay. In addition, to play this game, high system requirements will not be mandatory, because we are not going to use very high graphics. Thus, this game will be able to be played in every computer.

5.2 Graphical Smoothness/User Friendly Interface:

In computer games, animations and graphics are the core of the game for the users. Having a fluent gameplay, decent animations and graphics are some of the expectations of the users. For example, during the gameplay, when the user shoots or kills the enemy there should not be any irrelevant animation or graphical errors. Our aim is to create a game which can be favorable to be played by different ages. Hence we will create simple and clear UI so that all the users can be able to understand how to play the game. The gameplay and also the menus will be easily understandable.

5.3 Extendibility

Extendibility is also one of the main features that should be considered in the analysis report of a game project. New features and updates can be a good way to make a game be always up to date. Hence we have also thought in some new features we might implement in the future such as multiplayer feature, room chat, etc.

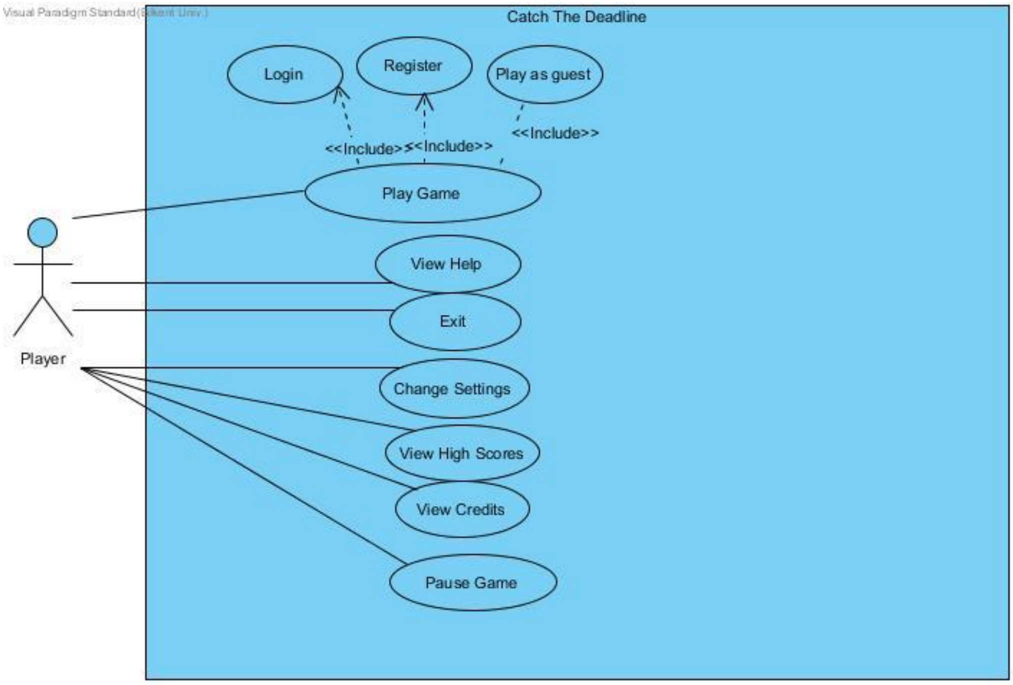
5.4 Additional Requirements

We don’t have non-functional additional requirement.

6 System Models

1.1 Use-Case Model

In the following diagram, we are providing the main cases that will happen in our game.



6.1.1 Play Game

Use case name:	*Play Game*
Primary actors:	Player
Entry condition:	Player is on the main page
Exit condition:	Player chooses “Exit” from Pause Game. Player successfully finished all of the levels. Player failed 3 times, therefore, has no life left.
Main Flow of Events:	1. The game starts 2. Choose the level, starting from the easiest one 3. Overcome the enemies by shooting at them with the appropriate weapon 4. Collect power-ups in order to keep playing the game 4. Aim to reach the end of the level within the given time 5. If the player successfully finishes a level, he/she gains access to the other levels 6. While the player passes the levels, the game gets harder because the number of enemies and obstacles increases 7. Player finishes all of the levels 8. Game displays a game over message 10. The score of the player is calculated by the system 9. Game displays the score 10. If the score is in between the highest 4 scores, modification in high score table is done 11. Game returns to the main page
Alternative Flow of Events	**1.** Player lost all of the life’s · Game is over · System continues from step 9 **2.** Player pauses the game · Game display the options in Pause Game Menu **3.** Player exits the game • Player pauses the game and choose exit from the Pause Game Menu • System continues from step 9

6.1.2 View Help

Use case name:	*View Help*
Primary actors:	Player
Entry condition:	• Player is on the main page • Player is in the Pause Game Menu
Exit condition:	· Player returns to the main page · Player returns to "Pause Game" page
Main Flow of Events:	1. Player pushes the help menu button 2. Information about game is displayed 3. Player reads the information 4. If player comes from the main menu, returns the main menu, if player comes from Pause Game Menu, returns to Pause Game menu
Alternative Flow of Events	**-**

5.1.3 Change Settings

Use case name:	*Change Settings*
Primary actors:	Player
Entry condition:	• Player is on the main page • Player is in the Pause Game Menu
Exit condition:	· Player returns to the main page · Player returns to "Pause Game" page

Main Flow of Events:	1. Player pushed the settings button 2. Game displays the settings 3. Player changes settings 4. System modify the game according to these settings 5. Player pushes back button and return back to main menu or pause game menu
Alternative Flow of Events	1. Player does not change the setting 2. System continues from step 5

6.1.4 View High Scores

Use case name:	*View High Scores*
Primary actors:	Player
Entry condition:	• Player is on the main page • Player is in the Pause Game Menu
Exit condition:	• Player returns to the main page • Player returns to "Pause Game" page
Main Flow of Events:	1. Player pushes the view high score button 2. Top 4 high scores and player names are displayed 3. Player reads the information 4. If player comes from the main menu, returns the main menu / If player comes from Pause Game Menu, returns to Pause Game menu
Alternative Flow of Events	-

6.1.5 View Credits

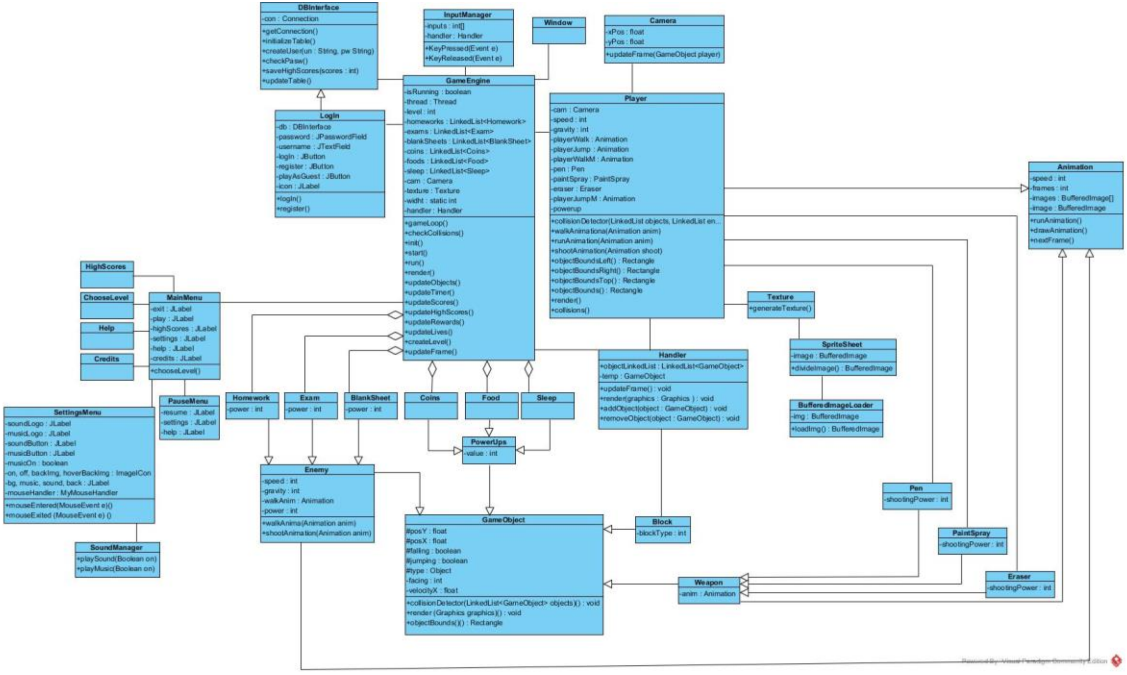
Use case name:	***View Credits***
Primary actors:	Player

Entry condition:	- Player is on the main page - Player is in the Pause Game Menu
Exit condition:	- Player returns to the main page - Player returns to “Pause Game” page
Main Flow of Events:	1. Player pushed the view credits button 2. Credits are displayed 3. Player reads the information 4. If player comes from the main menu, returns the main menu / If player comes from Pause Game Menu, returns to Pause Game Menu
Alternative Flow of Events	-

6.1.6 Exit

Use case name:	*Exit*
Primary actors:	Player
Entry condition:	- Player is on the main page - Player is in the Pause Game Menu
Exit condition:	-
Main Flow of Events:	1. Player pushed the exit button 2. Game displays “Are you sure you want to exit?” 3. Player select yes 4. Player exits
Alternative Flow of Events	1. Game displays “Are you sure you want to exit?” 2. Player select no 3. Player resume the game

6.2 Object and Class Model

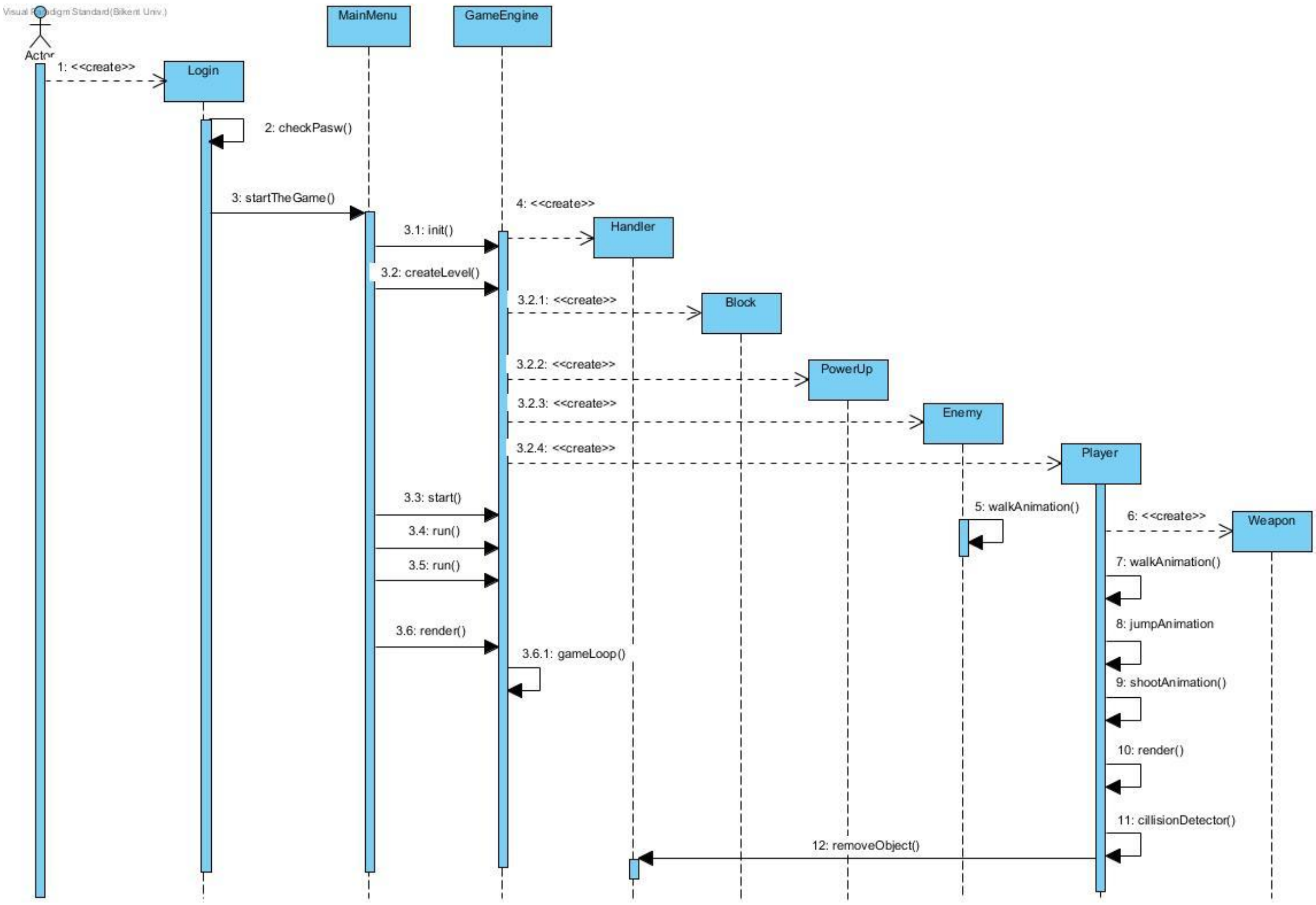


The first class that will be constructed in the main method will be the "LogIn" class and also the "DBInterface" class which will interact together in order to create the account of the user or to provide the possibility for the user to play as a guest. After the Log In is successful, this will trigger the "MainMenu" class to be constructed. The "InputManager" class deals with the interaction of the user with the game. "GameEngine" is the core class of this game. It will provide most of the methods and also will help in the interaction between all the other classes. This class will also deal with the drawing of the graphics. The "Animation" class deals with the animations of the game. The "Player" deals with the animation of player. The "Handler" class deals with event handling of the game. The "texture" class is responsible for rendering and loading the images for each animation.

6.3 Dynamic Models

6.3.1 Sequence Diagrams

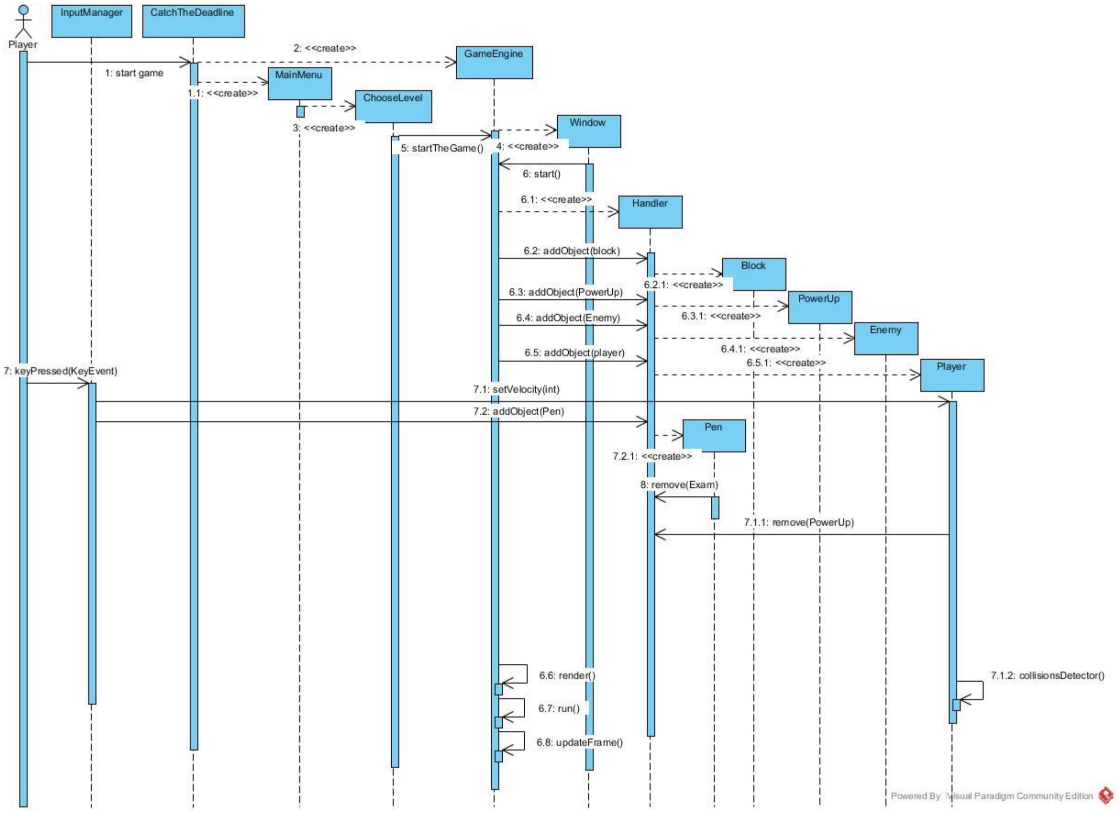
6.3.1.1 Start Game



Scenario: Start Game

Description: The user opens the game, he firsts sees the login screen and logs in with his account that he already made up earlier or he plays as a guest. After login he sees the main menu, where he clicks on play game and then the game starts. The GameEngine is the control object. The GameEngine creates the screen, initializes the UI, the frames and the instances of enemies, power-ups and player. The player initializes the weapons. The “gameEngine” runs in a loop so as to continuously update the game. Furthermore, if the player kills an enemy, it sends a message to the Handler class and it removes the killed object.

6.3.1.2 Play Game

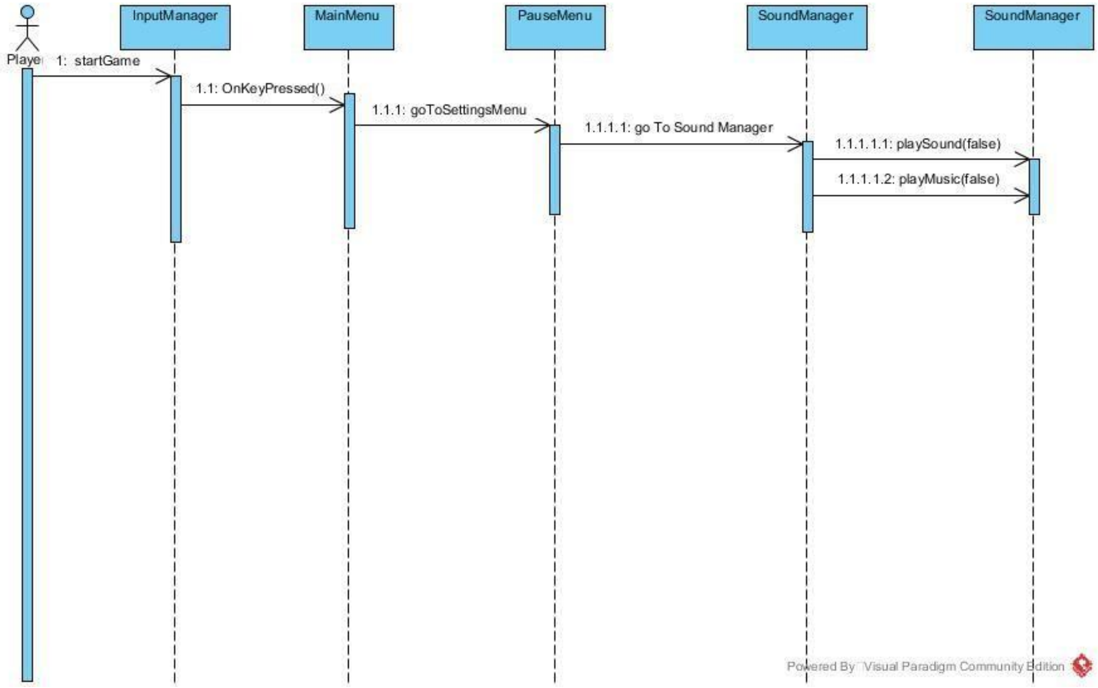


Scenario: Play Game

Description: Assuming the user has already logged into his account and selected Play Game option on the main menu and all the UI and instances of objects have been initialized, the player has the possibility to choose the level of the game. Then the “gameEngine” creates the level by creating all the objects of the game by adding them to the Handler class. After all objects have been created, the “GameEngine” enters a loop. The user makes the player move forward. Then the user sees a power-up of food and touches (collides) it, which in turn, increases the number of food items he had.

Then the player moves forward and encounters an exam (enemy). He shoots pen at the exams and the exam gets eliminated by the collision of pen and exam. The player moves forward but gets hit by an enemy of exam. Since he didn't react fast enough to eliminate it, he loses one of his lives. The player runs forward again and this time and he encounters exam which by using his weapon he eliminates it. After eliminating the exam, all the enemies that were supposed to be defeated in this level are defeated, hence the player reaches the end of this level and successfully submits his assignment on time. After this, his scores get saved into his profile and if they are among the highest scores they are updated in the high scores database.

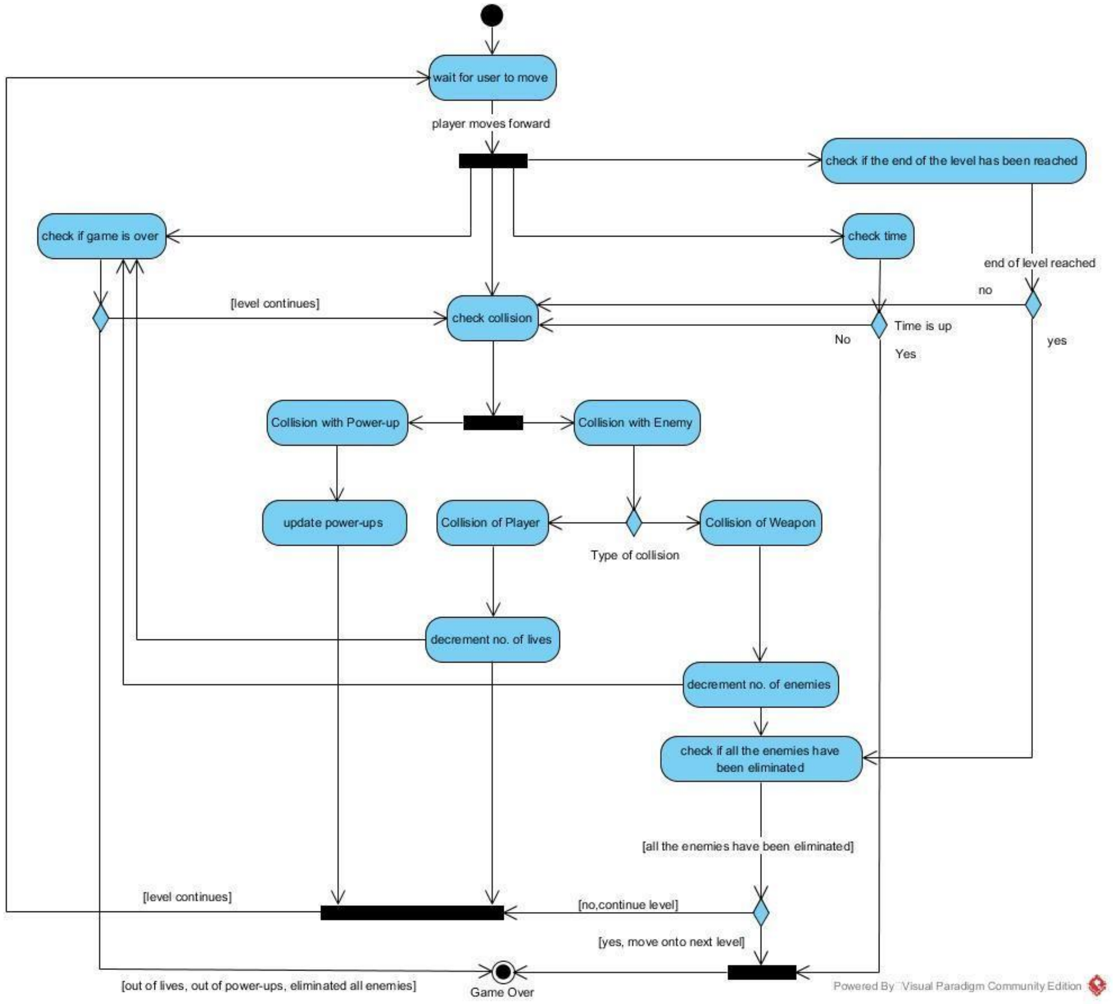
5.3.1.3 Change Settings


InputManager
MainMenu
PauseMenu
SoundManager
SoundManager
Playe
1: startGame
1.1: OnKeyPressed()
1.1.1: goToSettingsMenu
1.1.1.1: go To Sound Manager
1.1.1.1.1: playSound(false)
1.1.1.1.2: playMusic(false)
Powered By Visual Paradigm Community Edition

Scenario: Change Settings

Description: Here the player has already logged into his account and he wants to change the settings, so he clicks the settings option on the main menu. After entering the settings, which has options for turning music and sound on or off, the user decides to turn off the music and the sound. As he returns back to the main menu, the settings get applied.

6.3.2 Activity Diagram

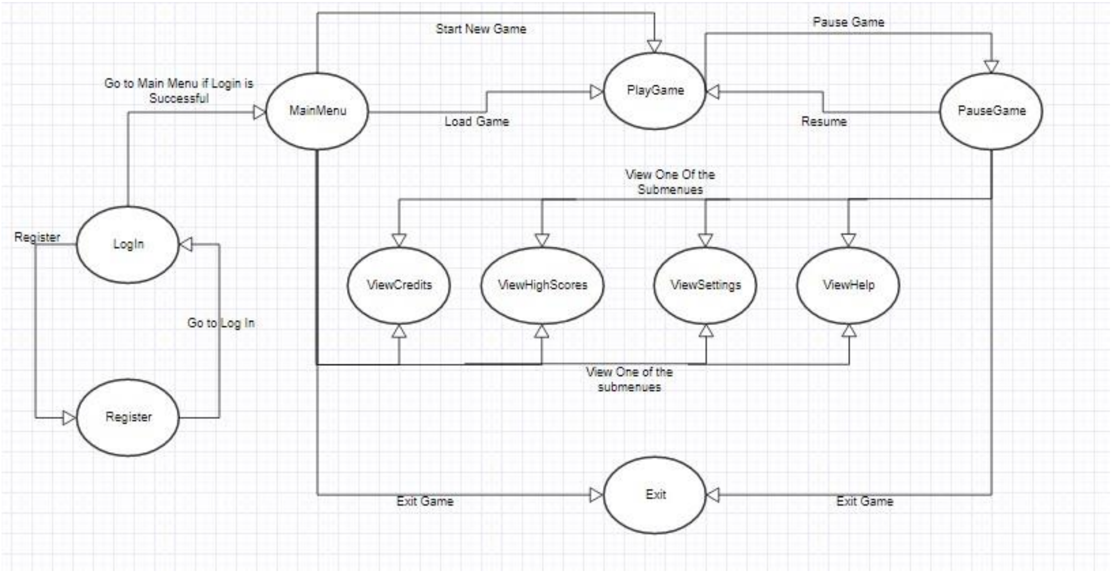


Description:

When the user starts to play the game, the player waits for the user to move forward. Then as the player moves forward, the system keeps on checking the collisions, the remaining time, the end of the level and other conditions such as running out of lives, out of power-ups, or out of enemies, that end the game or the level. If the player collides with a power-up then the power-ups will be incremented. If the player collides with an enemy the player will lose one of his/her lives, otherwise, if the weapon of the player has collided with the enemy, then the number of enemies supposed to get killed during that level will decrement. If all the enemies that are supposed to get killed during a level have been eliminated during the given amount of time, then that level will reach its end and the player move onto the next level. After this, if none of the conditions for when the game is over are met then the system will go back to its first step which is, wait for the player to move.

6.4 User Interface

6.4.1 Navigational Path



Register: Player should register if he/she doesn’t have an account.

Log in: The player who has an account should log in to continue the game. Then will proceed to the Main Menu

Main Menu: This window will show all possible options that the user can choose, such as starting the game, changing settings, viewing help, viewing credits or exit the game. Play Game: After the user presses the Play Game button, the system will load the game and the game will start.

Pause Game: Pause Game button provides to the user the pause game window which also contains all the options of the Main Menu.

View Credits: View Credits button can be pressed to see the information of game developers.

View Help: When view help button is pressed, system will show an information page, which includes the information about how to play the game.

Change Settings: Change settings button can be reached by both Pause Game and main menu. User can push this button if she/he wants to change the music or sounds options. View High

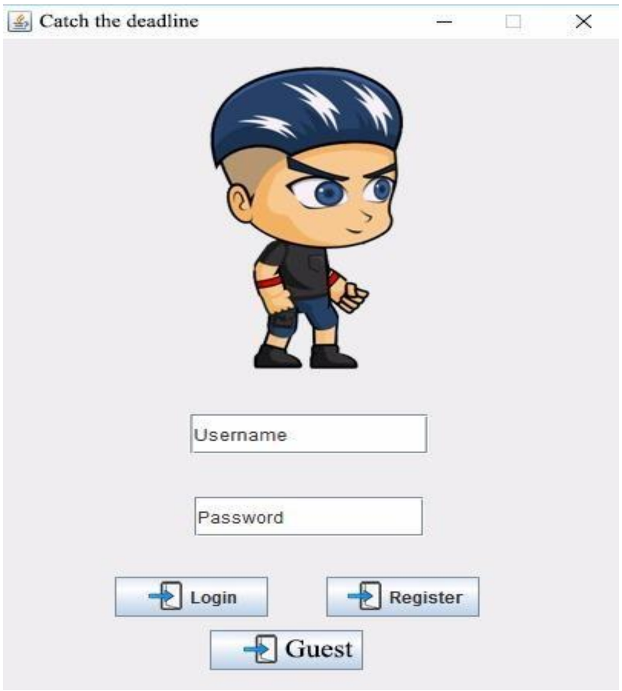
Scores: Player can hit View High Scores button to see the highest 4 scores and users who scored them.

Exit: This button can be used for quitting the game.

6.4.2 Screen Mockups

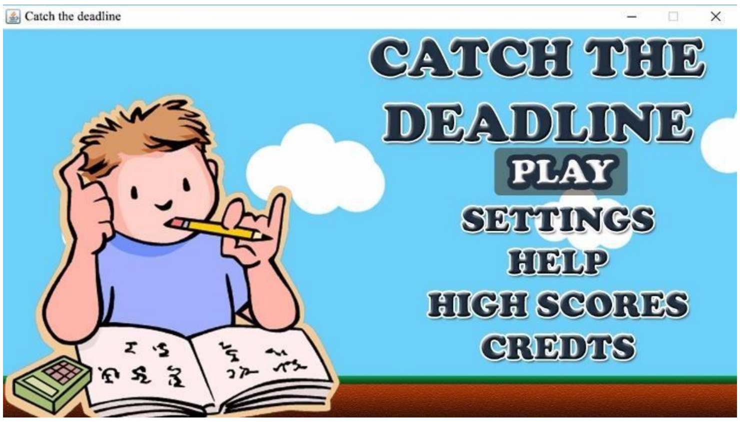
6.4.2.1 Login Screen

When the game starts, the first screen that will show up is login-register screen. Most of the games that are played today ask to create an account or play with your existing account. In our game we will also let the user play as a guest. If the player wants to play with his/her own account they must fill the required parts.



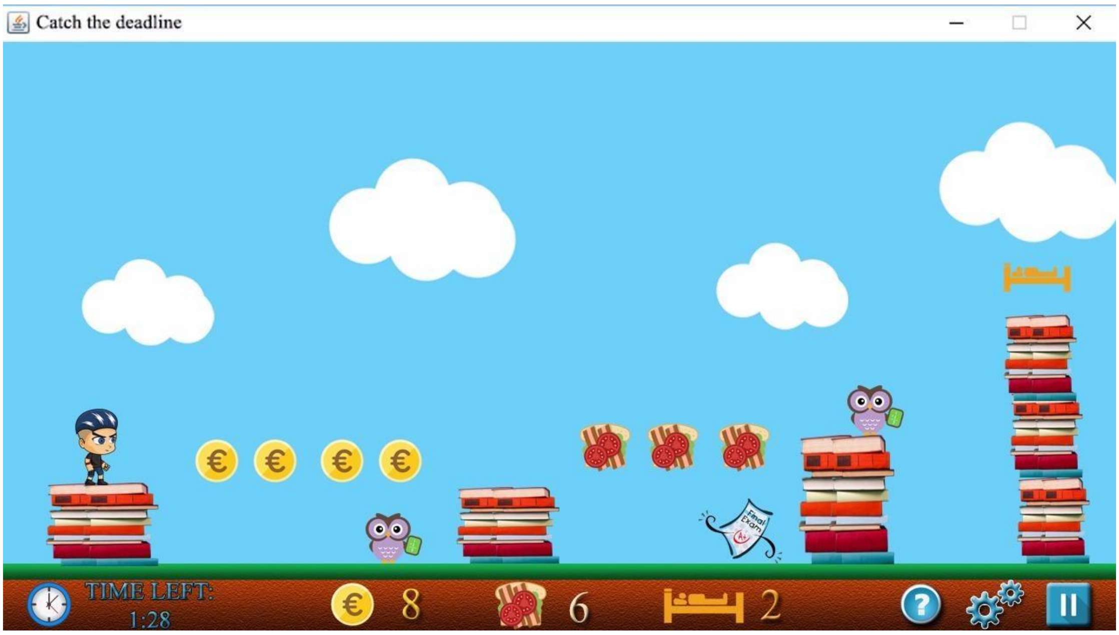
6.4.2.2 Main Menu

After logging in the game with your account or playing as a guest, player will encounter with this main menu. In our main menu, we will have several options like, play the game, go to settings, help menu, view high scores and credits part.



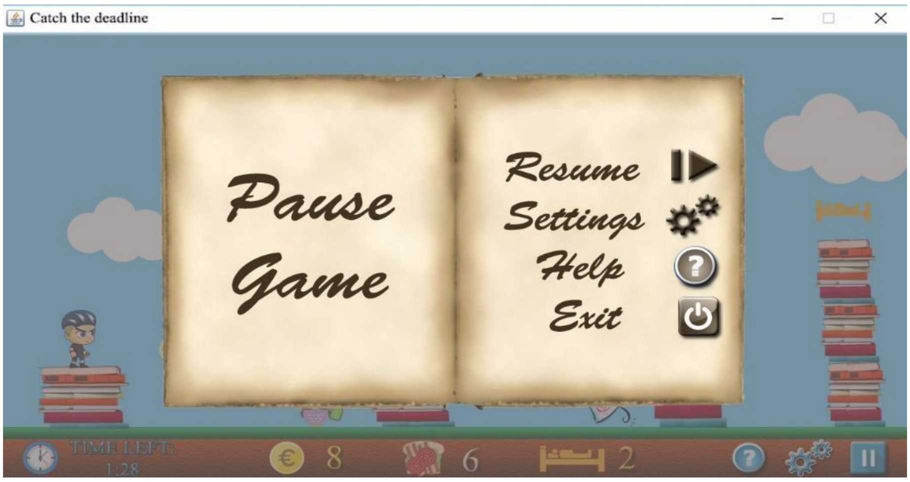
6.4.2.3 Gameplay

The gameplay screen is facing us with a colorful, friendly screen. Gameplay screen consist of our character, obstacles, enemies, rewards and weapons of the character. Each of these elements is selected so that it looks as user friendly as possible. Also the background of game remembers us the Mario Game that we used to play in our childhood.



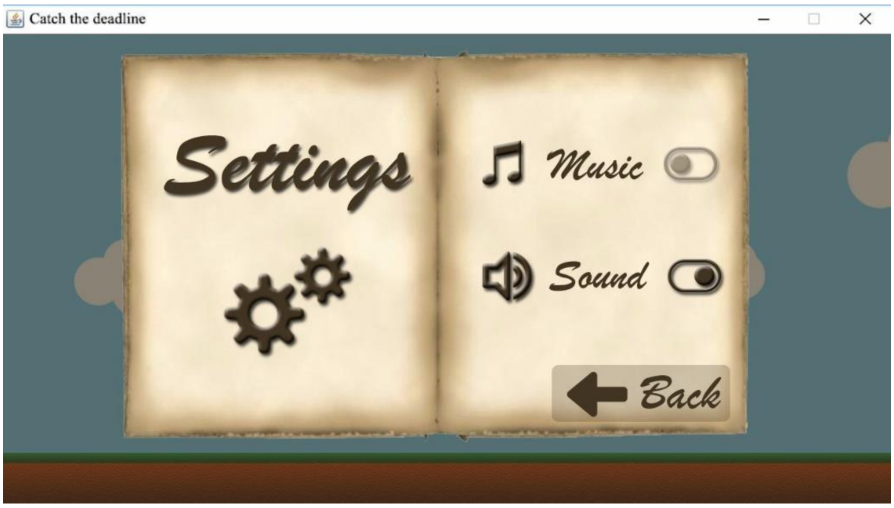
6.4.2.4 Pause Menu

Our pause game menu basically stops the game at that moment. This menu's background is going look like an opened book, and there will we be few choices that are provided to the user. Resume game, as it is well known, just lets the user keep on playing. Settings button provides to the user to go through the settings. In help button, a guide screen will appear and in exit button it stops the gameplay and lets the user quit the game after a dialog box appears to confirm the action.



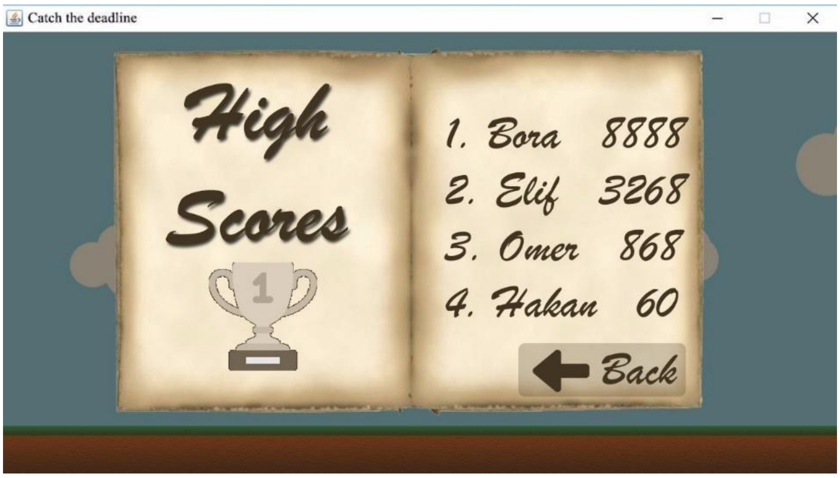
6.4.2.5 Settings Menu

In our settings menu, player can choose to turn on or off either music or the sound of the game. After the user changes his/her settings they can return to the pause menu or to the main menu, depending on where they came to this menu from.



6.4.2.6 High Scores

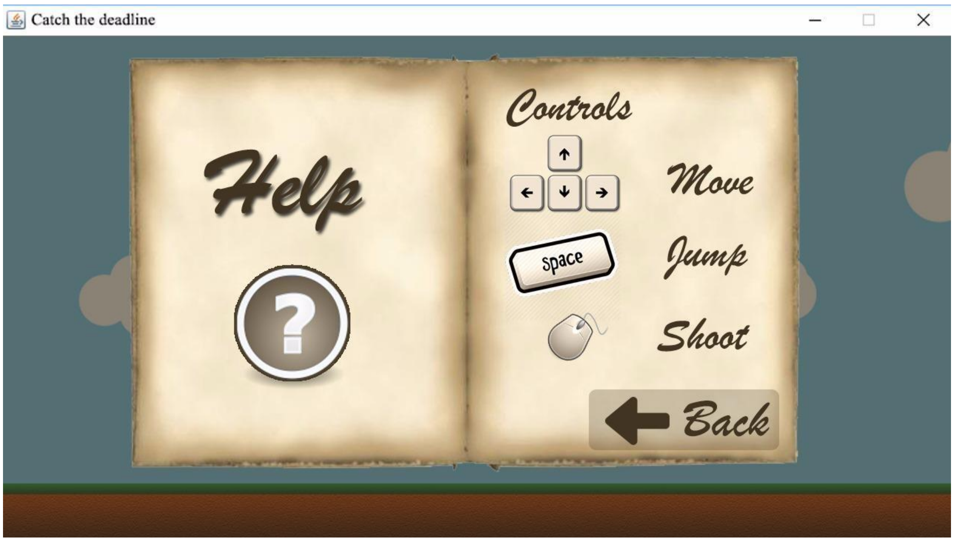
In the high score menu of our game, player basically can see the top rated players according to their scores. In this screen will be shown the 4 players that have got the highest scores. If a player passes one of the 4 highest rated users, this table will be updated.



1.	Bora	8888
2.	Elif	3268
3.	Omer	868
4.	Hakan	60

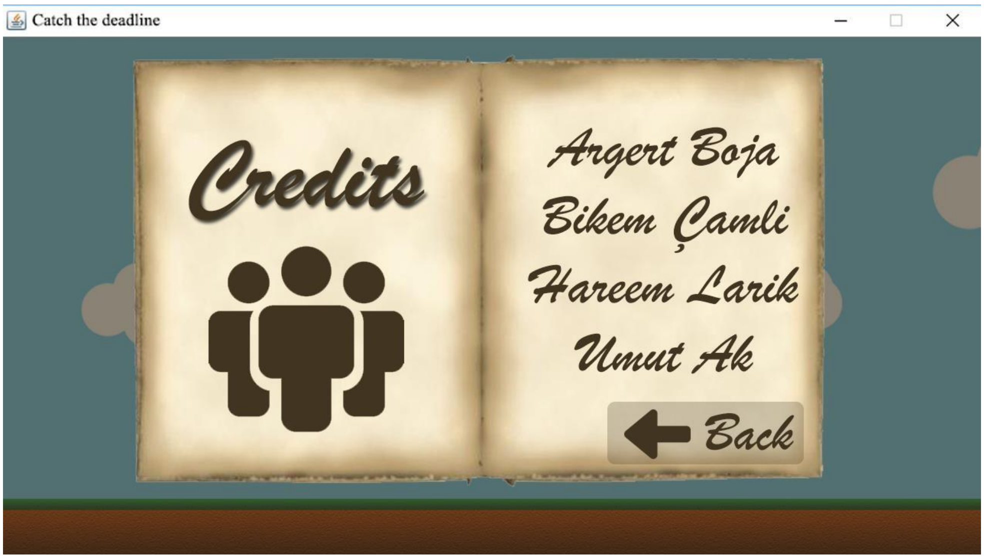
6.4.2.7 Help Menu

In the help menu, player will be encountered with the control buttons and they can be used. As in the other screens, user is free to go back to any other menus from help menu.



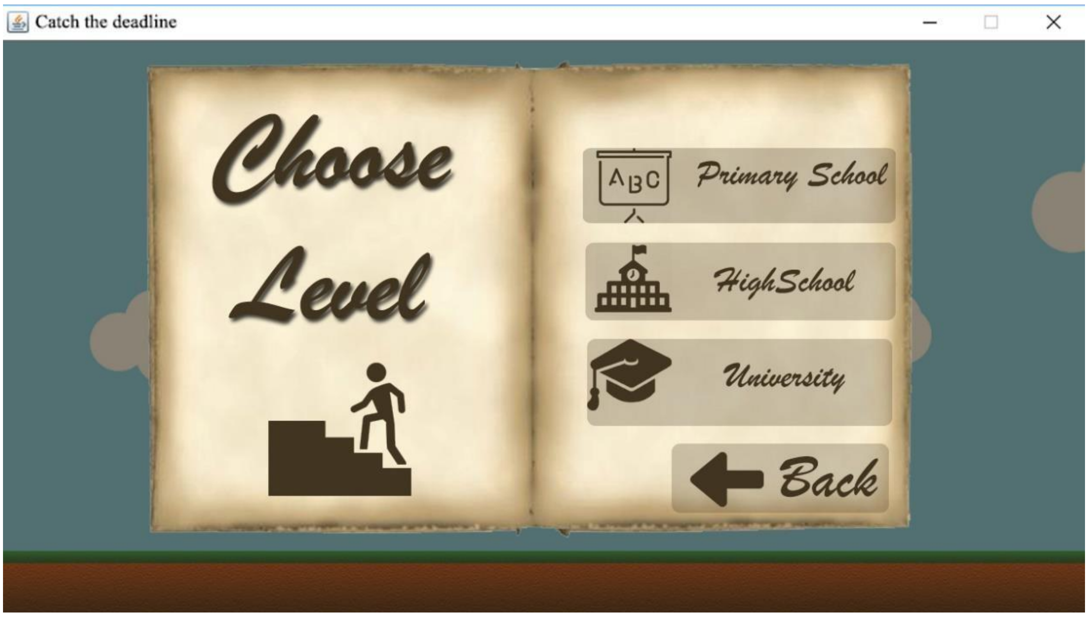
6.4.2.8 View Credits

In view credits menu player can see the developers of the game.



6.4.2.9 Choose Levels

Player can choose from 3 different levels (which has different difficulities) in choose levels menu. Primary school is the first and easiest level. High school is the second and medium level. University is the hardest and last level of the game.



7 Conclusion

Our analysis report aims to show the main idea and the way we are thinking on working for our project. As mentioned in the above paragraphs, our game will be a basic arcade game, consisting of rewards, enemies, weapons, etc. Furthermore, this report contains detailed information on the engineering methods and diagrams that we will use for this project. In this report we have also described the functional and nonfunctional requirements.

As it is known diagrams are crucial for the design of a program. Before creating the diagrams, we did some brain storming as a group. In this way we had a clearer idea of the diagrams that we had to design. Due to the limited time, we assigned a diagram for each member so that we could finish our report on time. As a group, we have allocated most of our working time on the diagrams, since the core of this analysis report is the models.

All in all, our analysis report has a big importance in creating a good project. This report will help us in the design report and furthermore it will definitely help us during our implementation.

8 References

[1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

[2] “Visual Paradigm”. https://www.visual-paradigm.com/ . Accessed: Oct 07, 2017.

[3] “Iconfinder”. https://www.iconfinder.com/ . Accessed: Oct 07, 2017.