Bilkent University

Department of Computer Engineering

Object Oriented Software Engineering Project

Crossit!

Final Report

Umutcan Aşutlu, Sarp Saatçıoğlu, Figali Taho, Utku Uçkun

Supervisor: Uğur Doğrusöz

Implementation report
Dec 24, 2016

1 Implementation Flow

The implementation phase of the project *Cross-it* is done by dividing the parts of the project among the group members. Two of us worked more focused on the controller and model classes, while the other two implemented a nice interface of the game. According to parts of MVC implantation pattern, we firstly did the basics of view package. During this process, basics of the GUI system were constructed. After that, we did the model classes of the game to finish-up with the all elements that will be used in the game. When it looked like all elements of our game were finished, we started to develop our controller classes which include all the mechanics of the game, such as manager classes and the game's engine.

During the implementation process, we have done most of the classes according to our pre-prepared design report. So, there are no major changes that happened during the implementation stage different than the report. One of the biggest change is we could not be able to finish all the power-ups that we desired. Other than that, all the other differences mainly occur at visual elements. For instance, the numbers of hats are less than which are planned before. In spite of the fact that the numbers of the some elements are not as much as we desired, every element is declared and built. The game mechanic of the Cross-it is provided. Object oriented design that our game stands for remains its qualifications.

There are also some new methods that we were not able to consider during the design stage. The methods that did not fulfill our services and didn't correspond to our ideas were changed with more appropriate ones and also new ones were added. For instance, there is changeHat() method now, that changes the model of the hat of the character for customizing.

2 User Guide

Cross-it is simple arcade game in which your aim should be to cross the road where vehicles pass. When you are able to cross all the lines of roads in the stage, the stage will end and the new stage, which is more difficult than previous one will appear.

2.1 Main Menu

In Main Menu, there are 6 options which you are able to select.

❏ New Game: Runs the game engine and starts a new game to play.

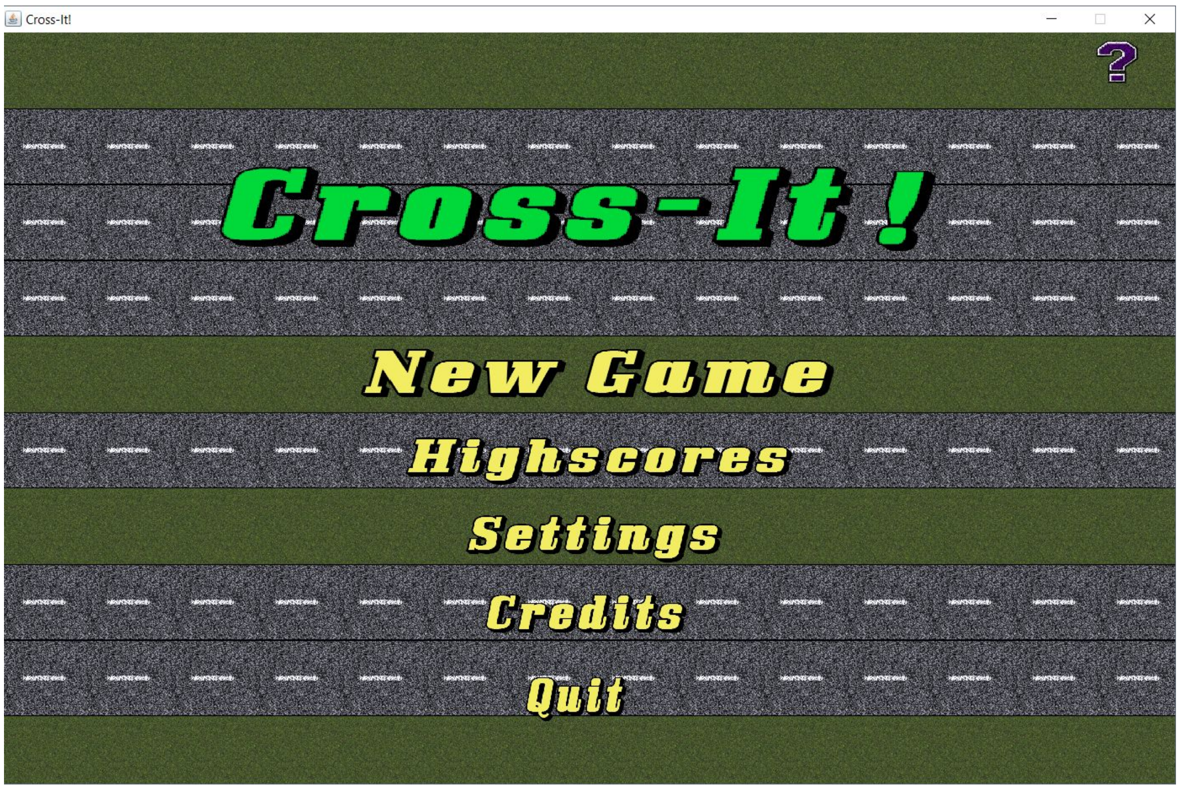
❏ Highscores: Shows the top 5 high scores if there is any, if not the page will be null.

❏ Settings: The segment for customizing the game by player's desires.

❏ Credits: Gives general out-game information such as developers' name.

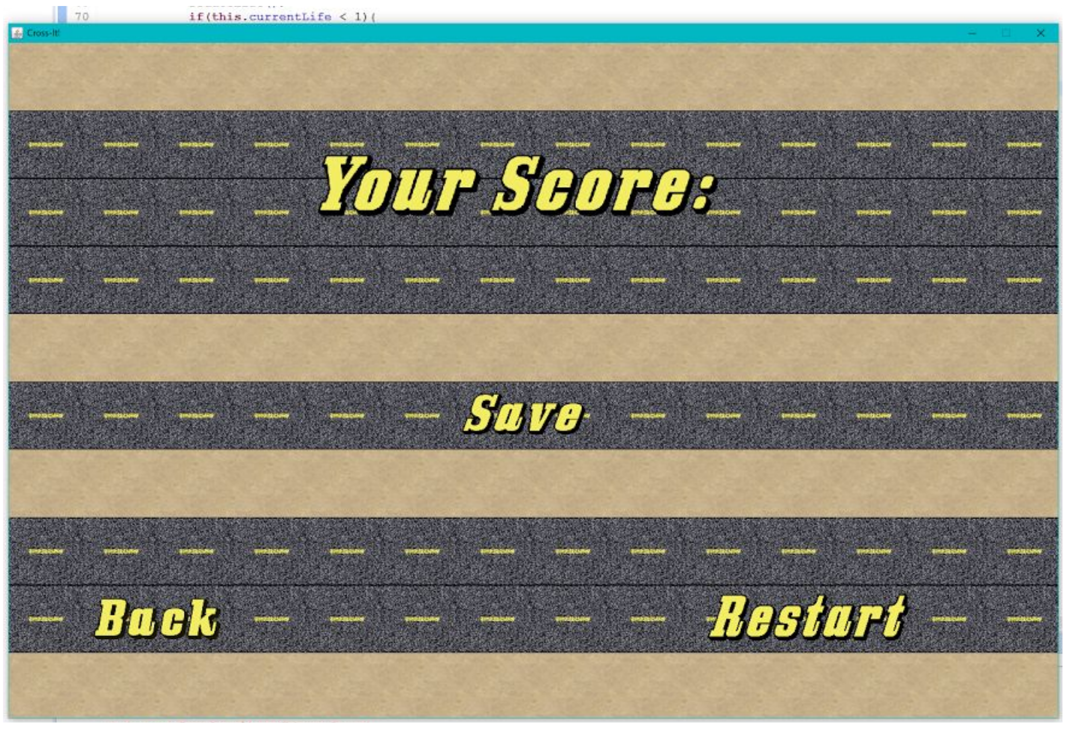
❏ Help (The question mark at top): Gives general information about the game.

❏ Quit: Closes the system and exits the game.



2.2 Highscores

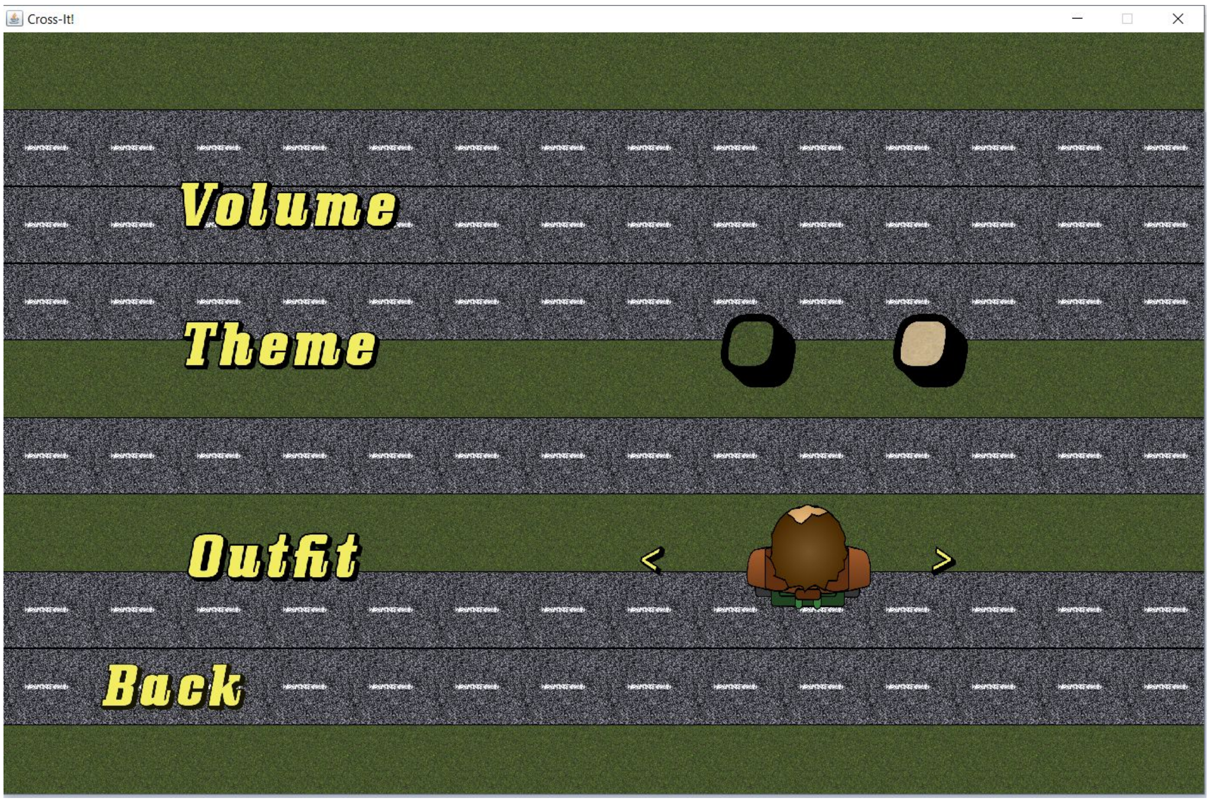
In Highscores, there will be top 5 high scores that recorded in any session of the game. If there are no high scores achieved yet, the menu will be like screenshot above. When the “Back” button is pressed, it will return to the main menu.

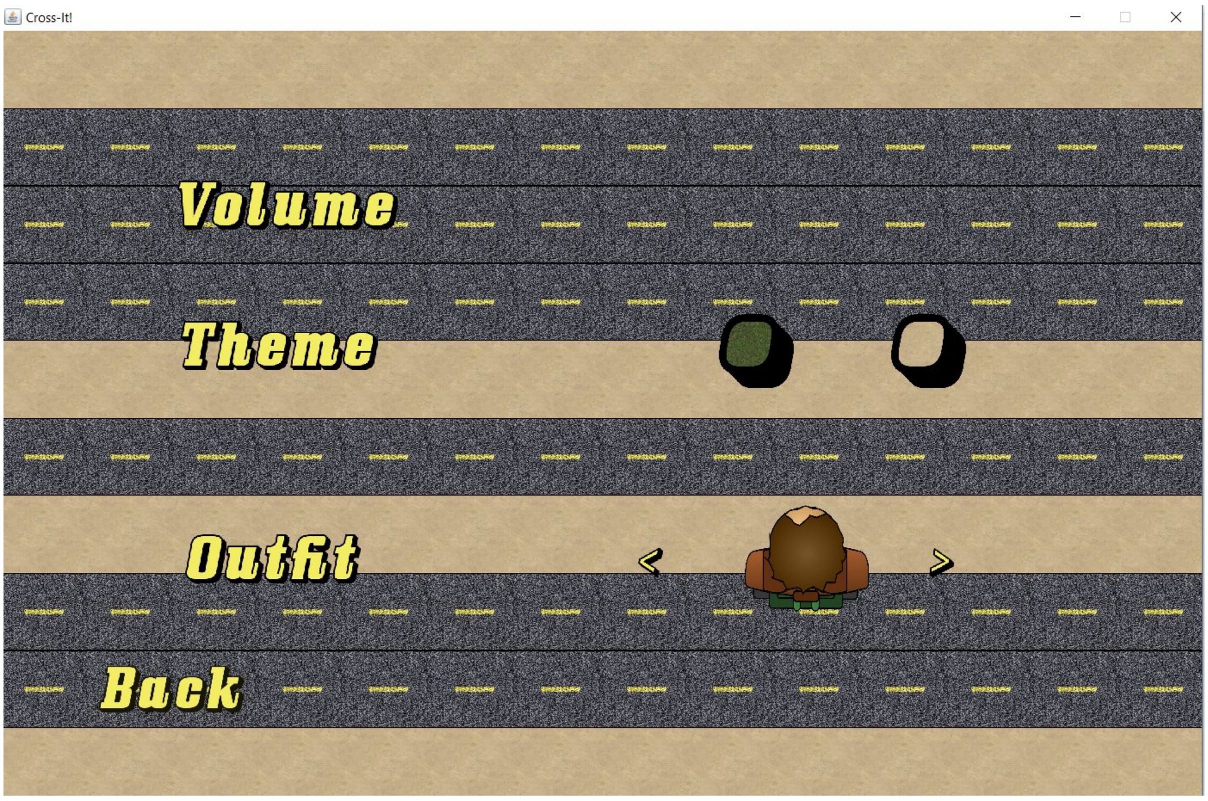


```
if(this.currentLife < 1){
```

2.3 Settings

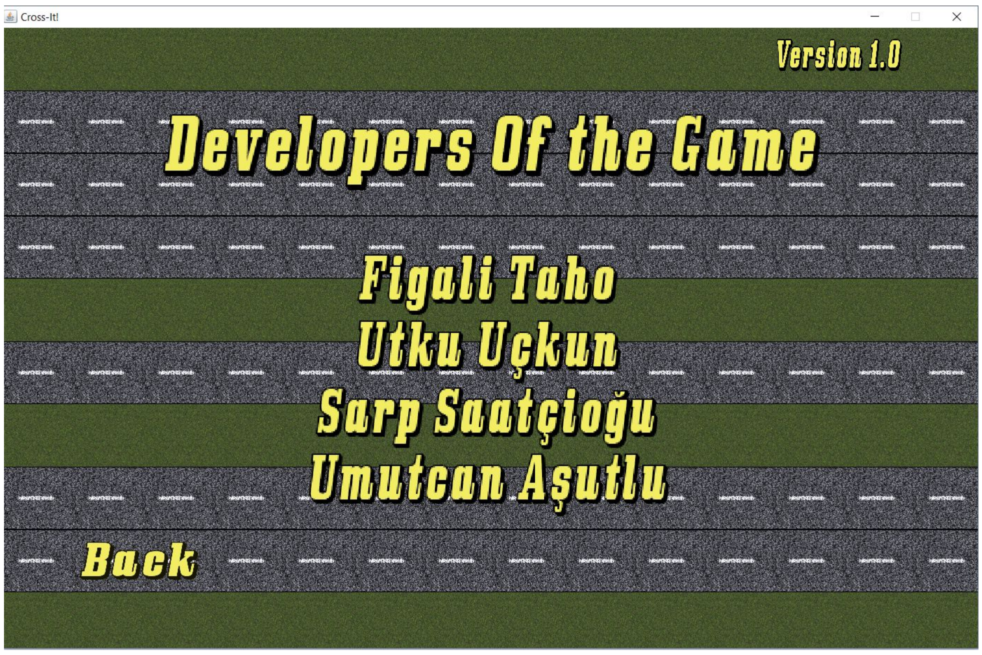
The settings menu is for customizing the game according to desires of the player. Player is able to change music volume, the theme of the game and the outfit of his character. There are 2 main themes for the game. First one is call Green Highway and the second one is called On the Sands. The buttons which are located on left and right on the character model changes the character model. The model is displayed is the current model in the game. Other models are purchasable in shop. When the “Back” button is pressed, it will return to the main menu.


Cross-It!
Volume
Theme
Outfit
<
>
Back


Cross-It!
Volume
Theme
Outfit
<
>
Back

2.4 Credits

The overall out-game information is given in this segment. The name of the developers and the version control information is written. When the “Back” button is pressed, it will return to the main menu.


Cross-It!

Version 1.0

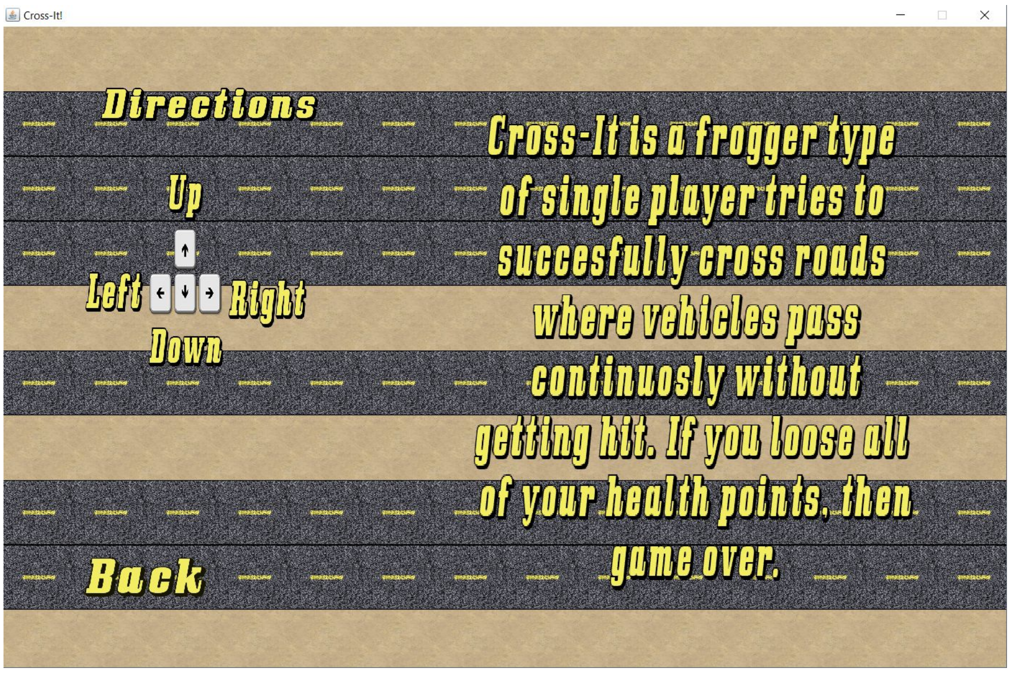
Developers Of the Game

Figali Taho
Utku Uçkun
Sarp Saatçioğu
Umutcan Aşutlu

Back

2.4 Help

The general game information and the hotkeys for controlling the character is given in Help. When the “Back” button is pressed, it will return to the main menu.



Cross-It is a frogger type of single player tries to succesfully cross roads where vehicles pass continuosly without getting hit. If you loose all of your health points, then game over.

3 Status of Implementation

Even though our game Cross-It is not implemented properly, almost all functional requirements are handled properly. The game application starts in a Main Menu and all options can be chosen as described in requirements. When the New Game button is clicked the game can start and it is playable in terms of moving character across the road when cars are passing through the road. Moreover, vehicles can have different directions as mentioned in requirements. The settings work properly in terms of changing the outfit(hat) of the character and changing the theme of the game as in requirements. Credits also shows the developers of the game as in the functional requirements. Help section is also working properly as we proposed in the requirements.

The collision check in the gameplay is properly implemented and it reduces life of the player as proposed in the requirements.All collectables and mystery boxes are handled, although with minor changes in the code due to limited time. The scoring in the game is currently handled, with every update of the game increasing this score.

A difficulty we had while implementing the game was dealing with the main thread of the game. Since the game depends on the vehicle speed, the user would work on a different thread and the vehicles would update separately. This happened mainly because we handled the case where the user wants to move faster than the vehicles since the vehicles are to move slowly in the initial stages. We developed a timer-like thread that would work like a speed measurer. The timer-thread of the game updates the screen and the character and all the vehicles move forward.

Overall, we completed many things in this project. Although we couldn’t complete all of the requirements we said to achieve, in the limited amount of time we created a game that has nothing but small bugs. Moreover, we learned a lot about threading and developing with using Git as a version control system. We had to deal with a lot of errors, and github mostly helped us since we could see our own errors, while we also used debuggers extensively.