CS 319 - Object-Oriented Software Engineering

System Design Report

Panzer 2017

Group 2-C

Pınar BAYATA

Burak SİBİRLİOĞLU

Ndriçim RRAPİ

Ferhat Serdar ATALAY

Contents

1. Changes In the Implementation

A. Speed(Additional Feature)

In our design report, we thought that the speed will be constant during the game. However, we decided during the implementation that the speed will increase if playerTank reaches to a bonus such as the speed will increase if playerTank gets a speed bonus.

B. Tank Class

In the tank class, we said previously in our reports that there will be one move() method. However, fort he ease of implementation, we divided the method into four parts such as moveLeft(), moveRight(), moveDown() and move Up().

Apart from the methods, we changed the parameters of Tank class. We added ArrayList<Image> _icon in order to have different images of tanks. For example, if the tank goes left we will have a left looking tank image and it will be the sam efor other directions.

C. Main Class

We defined a new class which is called Main Class initializing the Main Menu.

D. Buttons

We changed our buttons in order to implement it easily. The buttons are 'A' for the left direction, 'S' fort he down direction, 'D' fort he right direction and 'W' for the up direction.

E. JavaFX

We used JavaFX to replace swing as the standard GUI library for JavaSE.

2.Status of the Project

We have been implementing our Panzer17 game. We are trying to be consistent and we are implementing our game with the help of our object class diagram in Design Report. There is a main screen , the player can choose whether to play or to see the HighScores, Credits and so forth. When the player chooses the 'Play' button, a tank and a map will appear on the screen. The tank moves in the up, right, left and down directions perfectly but we are also working on the bullets and bonuses. We identified the size of the bricks, to fit in to the screen and bricks' size will be important for the tank. The player can view Credits also. The screenshots are in the following sections.

3.User's Guide

3.1. System Requirements

Panzer17 is a game which implemented in Java. We used JavaFX to replace swing as the standard GUI library for JavaSE[1]. Therefore, users should download JavaSE.

Minimum System Requirements:

- Windows XP or Mac OS X Snow Leopard
- Pentium2 233 MHz CPU or higher
- 256 MB of RAM or higher
- Screen resolution: 800x600

Recommended System Requirements:

- Windows 10 or macOS Sierra
- Intel i5 2 GHz CPU or higher
- 1 GB of RAM or higher
- Screen Resolution: 800 x 600

3.2. Installation

If the users know how to implement on Java, the game can be downloaded from our Github page which is https://github.com/buraksibirlioglu/CS319_SEC2_2C. NetBeans can be used as a Java program.

3.3.Overview of the Game

When the player presses play button , Panzer 2017 starts with the top view of a map and tanks. The map consists of various bricks such as one brick can be collapsed with one shoot while the other one should be shot three times. There are two castles, one for the enemies and one for the player. The castles are surrounded by the easily destroyable bricks. There will be 3 levels and in each level, map's design is changed so that it will be hard for the player to move and catch an enemy. Enemies are classified in themselves. Like the bricks, some of them have to be shot more than the other ones. Heart counter starts to count from 3 and decreases the value whenever player is shot by an enemy. Apart from that, point counter will start from 0 and adds the points as the player shoots enemies. An enemy will not be able to shoot other enemy. An enemy's tank will be different from the player's. Enemy's position will be generated randomly in every level. The goal is to play all the levels and reach the castle in every levels.

3.4. Game Objects Management Subsystem

3.4.1.Objects

Player Tank: Player tank is our main object which should avoid from enemies and should protect the castle.

Bullet: When player presses space button on keyboard, there will be a bullet.

Enemy Tank: In all levels, there will be enemies.

Bricks: Bricks divide into four brick types such as green brick, red brick, blue brick, white brick.

Bonus: There will be bonus objects that will appear randomly in the map.

Castle: There will be two castles, one for enemies an done for player.

3.4.2. Controls

'A' Button: Go left

'S' Button: Go down

'D' Button: Go right

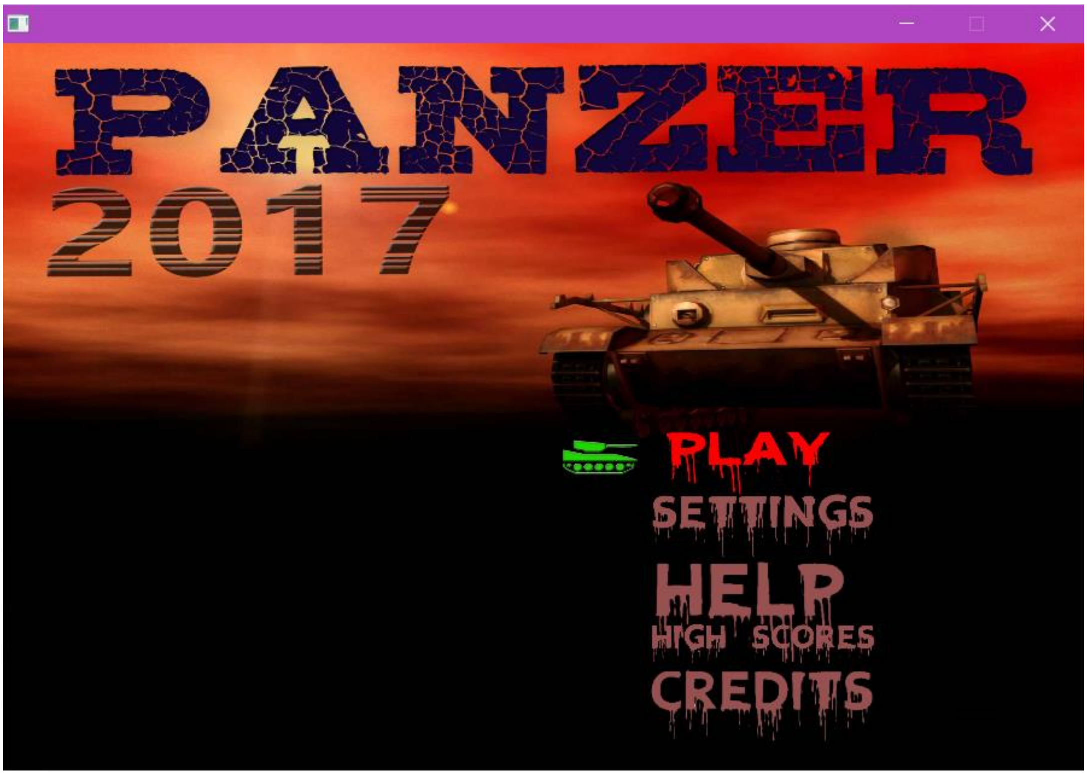
'W' Button: Go up

Space: Fire

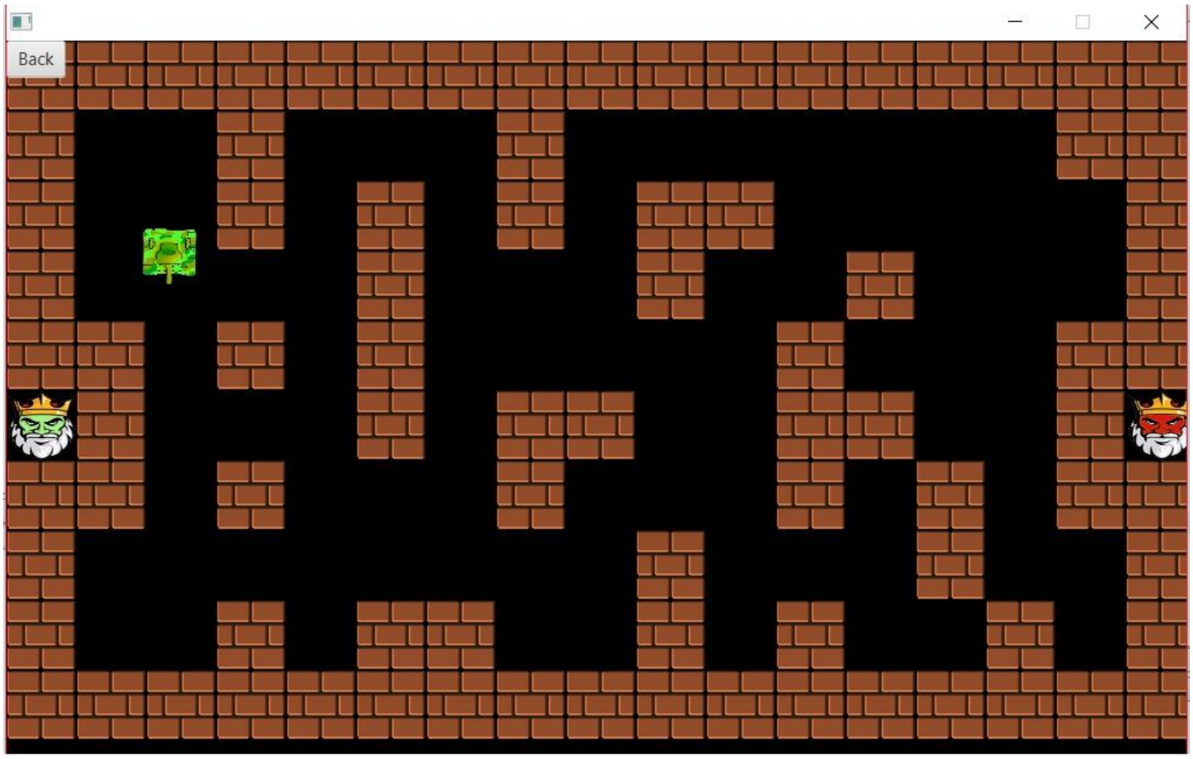
3.4.3. Game Screenshots & Menus

The Game Screenshots shows the status of the project. Because the implementation is not over yet, there is no 'High Scores' part.

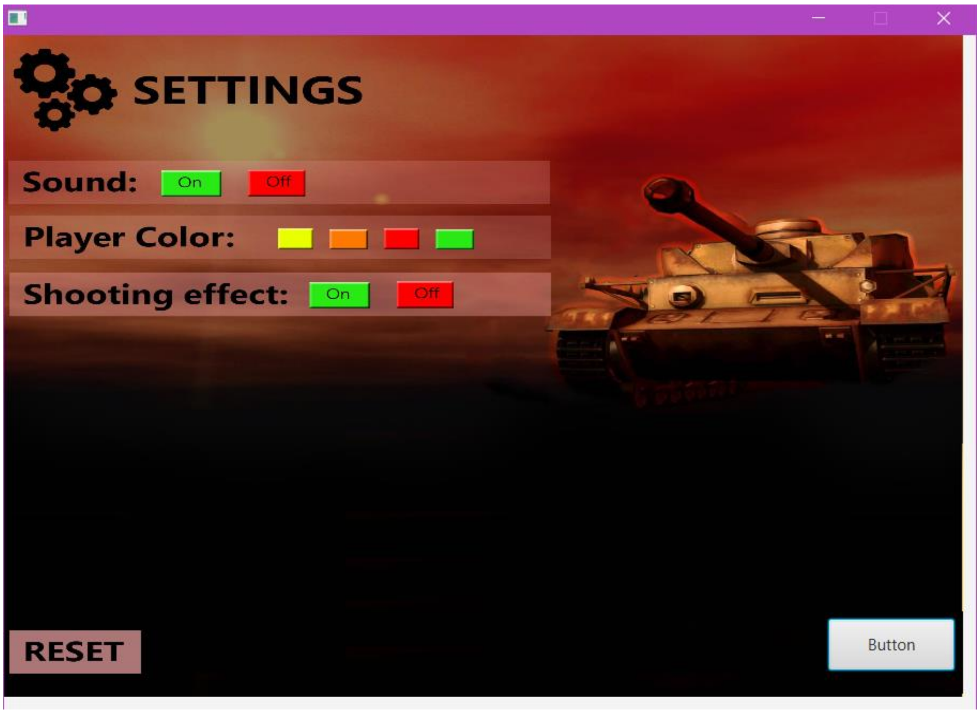
Main Menu: Opening screen is main menu and it contains 4 options: Play, Settings, Help, High Scores, Credits.



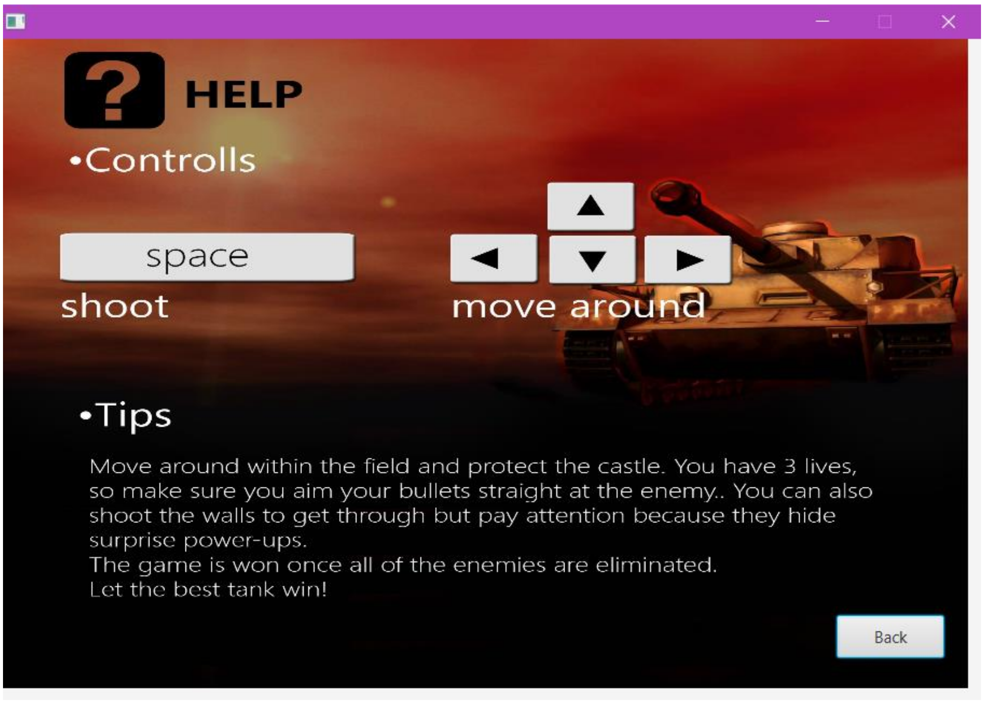
Play: If player chooses play button, the game starts.



Settings: Player can choose Settings and the screen below shows up.

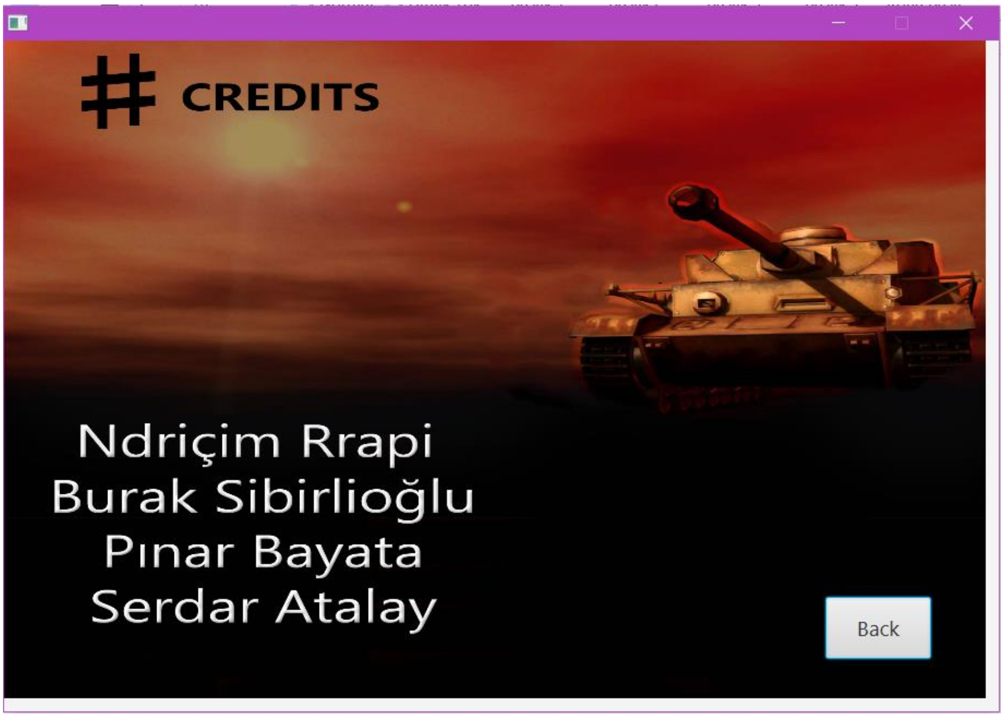


Help: If this option is selected, player can see the help instructions.



Move around within the field and protect the castle. You have 3 lives, so make sure you aim your bullets straight at the enemy.. You can also shoot the walls to get through but pay attention because they hide surprise power-ups.
The game is won once all of the enemies are eliminated.
Let the best tank win!

Credits: User can view the implementers by selecting ‘Credits’



Ndriçim Rrapi
Burak Sibirlioğlu
Pınar Bayata
Serdar Atalay

4.References

[1] https://docs.oracle.com/javafx/2/overview/jfxpub-overview.html