CS 319 TERM PROJECT

Section 1 Group 1I

Risk Game Final Report

Emin Adem Buran - 21703279

Onur Oruç - 21702381

Ömer Yavuz Öztürk - 21803565

Melike Demirci - 21702346

Yusuf Ziya Özgül - 21703158

Supervisor: Eray Tüzün

2.3 Usage of Design Pattern 3

2.4 Designing UML Diagrams 3

2.5 Usage of Github 3

2.6 Usage of Java FX 4

2.7 Creating Executable 4

2.8 Writing Analysis and Design Reports 4

3.User's Guide 4

3.1 Main Menu 4

3.2 Initializing New Game 5

3.3 Options 6

3.4. Stages In the Game 6

3.4.1. Buy Stage 7

3.4.2. Draft Stage 7

3.4.3. Attack Stage 8

3.4.4. Fortify Stage 9

4. Build Instructions 10

5. Work Allocation 10

1.Introduction

Our implementation process started after we have finished the design report. We have decided to use IntelliJ IDEA to implement our project. Each of us downloaded it and connected to it our remote project repository CS-319-PROJECT-1I-RISK. Moreover, we have used open JavaFX which is included in Liberica jdk.

Core game components are now completed; four different stages Buy, Draft, Attack and Fortify can be done. Also the game can be saved and loaded. We couldn't test the plague and weather condition modes but the other additional features such as buy mercenary, gold mine, commander, events and motivation of troops are implemented successfully.

2.Lessons Learnt

2.1 Team Work

One of the important lessons we learned from this project is that we learned working as a team. Teamwork is related to the distribution of roles. We learned distribute the roles among us for both implementation of the project and writing analysis and iteration reports of our project.

2.2 Code Review

After completing this project, we learned how to review our codes in different classes written by teammates and learn how to write code in other classes depending on the code of our teammates.

2.3 Usage of Design Pattern

One of the important contributions of this project to us is we learnt to use design patterns in the project. We have used Façade, Singleton and Strategy Design patterns.

2.4 Designing UML Diagrams

We designed our diagrams in the analysis and design iterations report to determine a way to how to implement our project. In the implementation stage, we wrote codes and classes depending on these UML diagrams.

2.5 Usage of Github

One of the most important contributions of this project is that we learn how to use Github effectively. We learned how to create our own branch, how to pull the current version of the project to the local repository, how to commit our changes and push the codes we

implemented to Github by using SourceTree and Intellij, and how to solve conflicts and merge different branches.

2.6 Usage of Java FX

When we were handling GUI management of our project, we used Java FX for this. Because it is very powerful for front-end and includes more advanced properties rather than other Java libraries for GUI management.

2.7 Creating Executable

We needed to write executable to run our project in the project demo presentation. So, we created an executable for this. It is beneficial because we can use the same logic for future projects.

2.8 Writing Analysis and Design Reports

The important part of our project is writing analysis and design reports before the implementation stage. We learned how to write them carefully.

3.User's Guide

3.1 Main Menu

Figure 1 : Main Menu

When the player(s) executes the game, the player(s) see(s) the main menu. Main menu consists of 6 main buttons. Player(s) can start a new a game, continue to a loaded game, set the options, see the rules, see the credits, and exit the game.

3.2 Initializing New Game

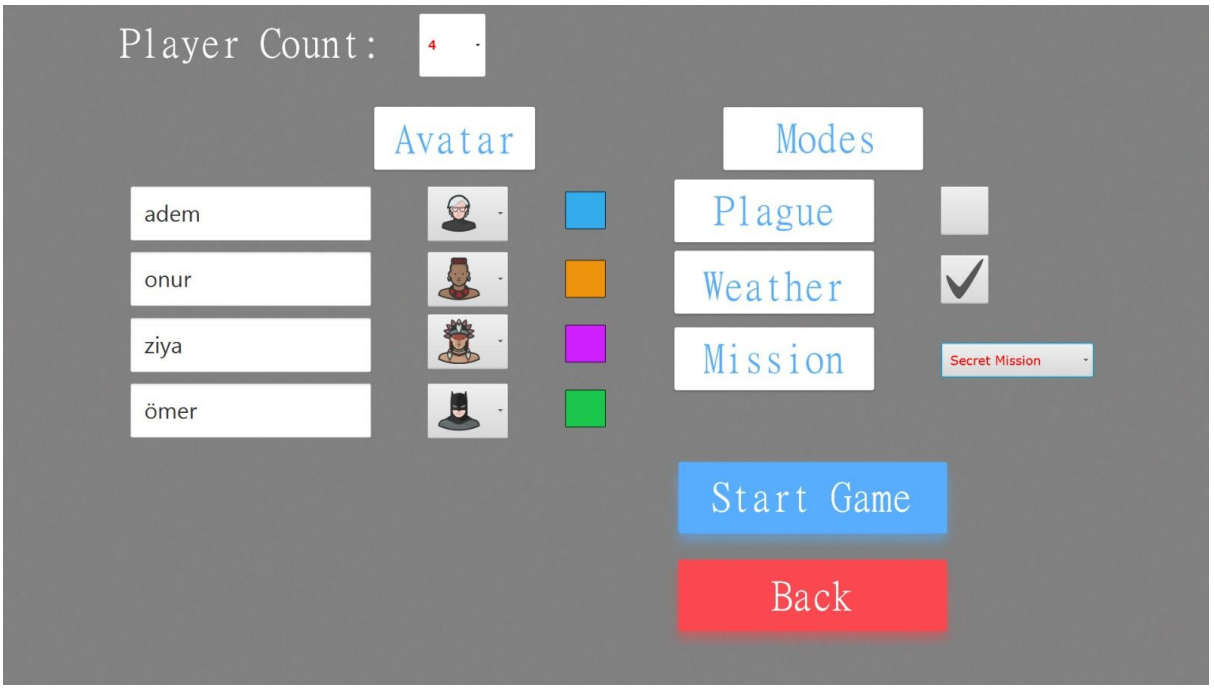

Figure 2 : Creating new game and start

Players can choose the number of player count. There are 3 options, the game can be played by 2 or 3 or 4 players. When player count is selected, the player will enter their names and select their avatars. Players will select modes for the game. There are 3 modes. If players select plague mode, there will be a plague in some regions in the game randomly, and it will change dynamically until the game finishes. There are two types of missions in the game : Secret Mission and Global Domination. Players will select one of them. After players select modes and enter their information, players can start the game by pressing the start game button. Another option is that if players change their decisions, they can back to the main menu.

3.3 Options

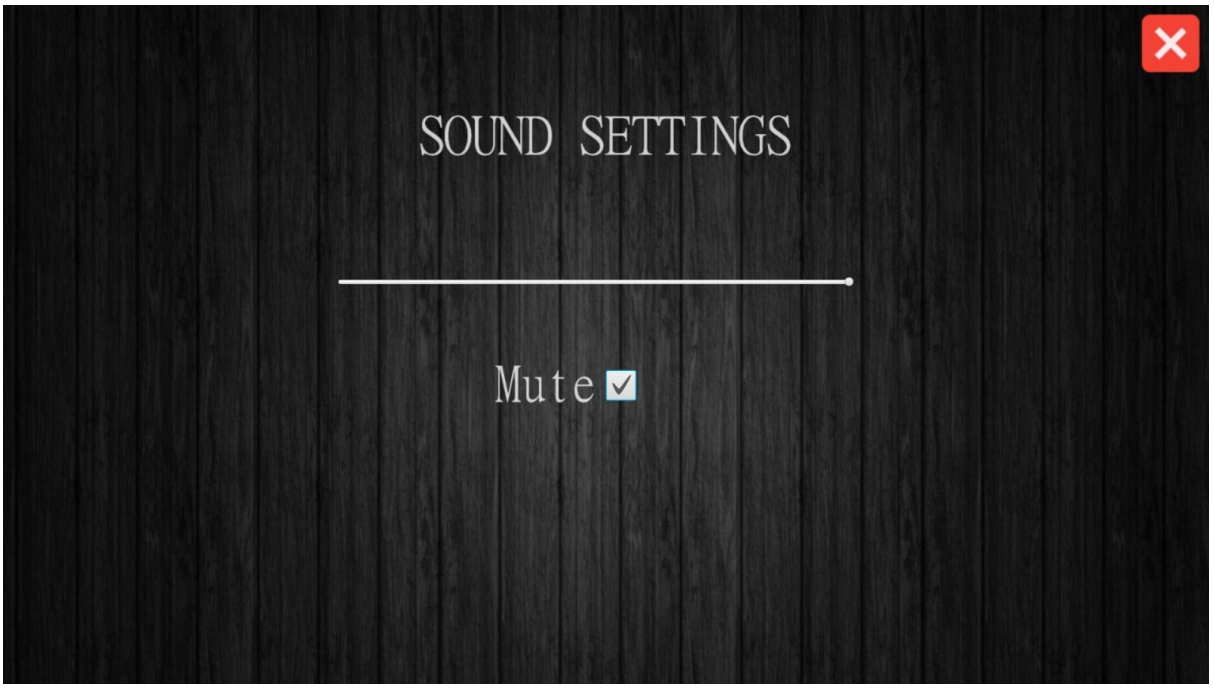

Figure 3 : One screenshoot from options menu.

Players can adjust the volume in the options menu. Players can mute the volume by using the checkbox next to the mute. Players can also increase or decrease the volume using the line above the mute option.

3.4. Stages In the Game

At the beginning of the game, the game map should be initialized by clicking on the distribute regions button.

3.4.1. Buy Stage

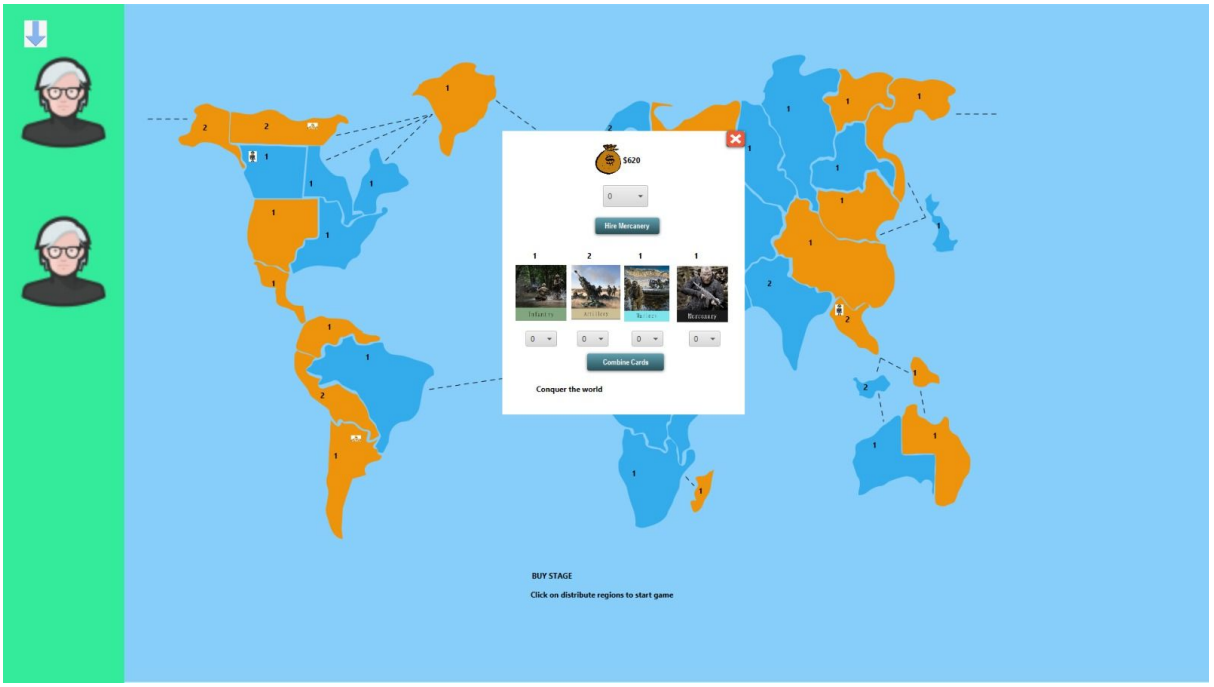

Figure 4 : Buy Stage

In the buy stage, players can buy mercenaries, combine troop cards, and organize event.

- In order to buy a mercenary, players should open the profile panel by clicking the player icon on the screen. On this player profile, the amount of money will be displayed. Players should select the number of troops to be bought and click on the buy mercenary button.
- In order to combine troop cards, players should open the same player panel and should select the number of cards to be combined.
- In order to organize an event, players should click on the organize event button on the game map and should select one of the owned regions.

Players can click on the next stage button without any condition.

3.4.2. Draft Stage

In the draft stage, players can draft their troops and commander.

- In order to draft troops, players should click on one of their regions and select the number of troops from the screen.
- In order to move commander, players should click on the move commander button and select one of their regions.

Players can click on the next stage button if they drafted all of the troops.

3.4.3. Attack Stage

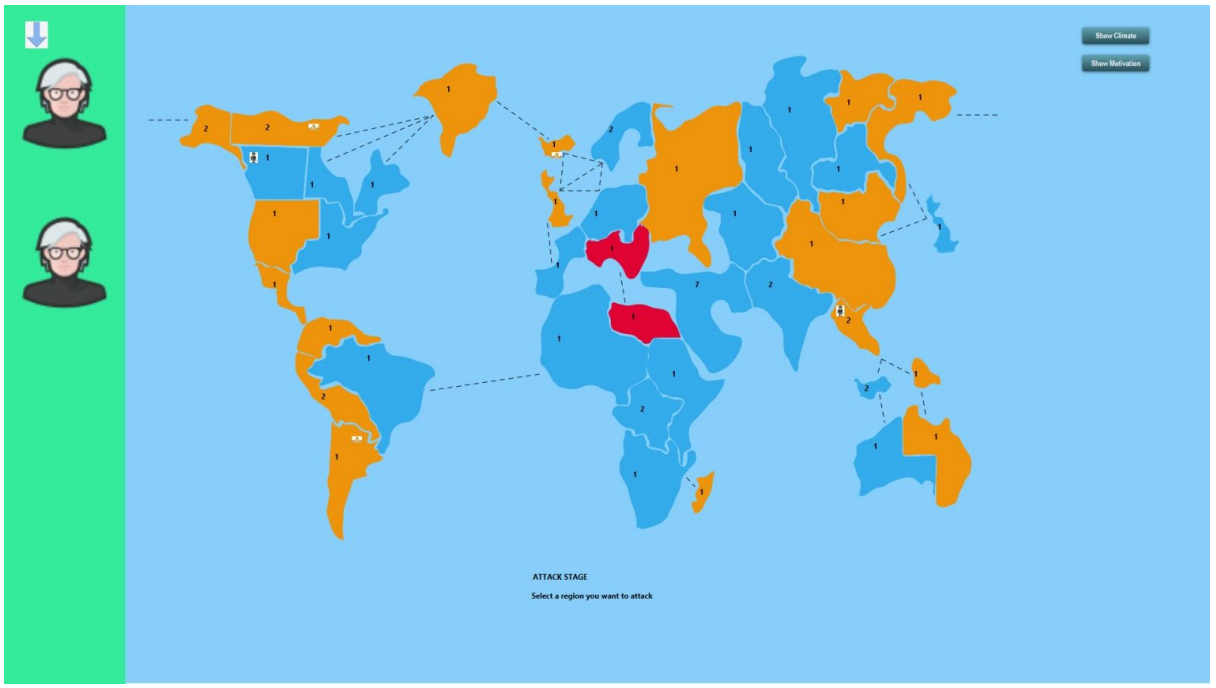

Figure 5: The region(s) that can be attacked from the selected region is colored as red.

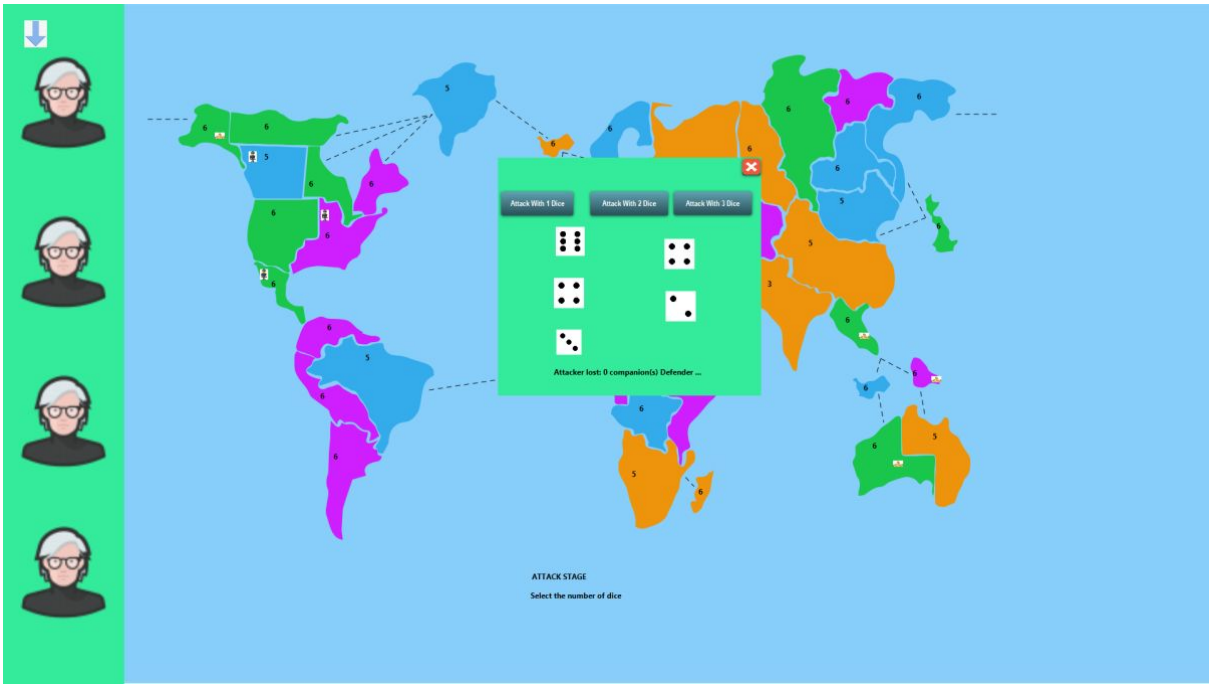

Figure 6 : Dice Panel

In the attack stage, players can organize an attack from their owned regions. In order to attack a region, players should select one of their regions that have at least 2 troops. After that map will display neighbor enemy regions in red. The player should select one of these regions to attack. When the attacked region is selected, the dice panel will be displayed. From this panel, players can choose the number of dice to be used while attacking. When the attacked region is conquered, players should select the number of troops to be invaded to the conquered region. Players should invade at least the amount of troops used in the last attack.

Players can organize attacks as much as they want if they have enough troops on their regions. Players can click on the next stage button without any condition.

3.4.4. Fortify Stage

In the fortify stage, players can change their troop's places. In order to fortify a troop from one region to another, players should first click on one of their regions which will send troops, and then selects the number of troops to be fortified. Lastly, players should select the region which will receive troops.

Players can click on the next stage button without any condition.

4. Build Instructions

Jar file of the game can be accessed from the Github page to run the game directly. Thus, the source code of the game is accessible in the Github page as well.

The first source code should be cloned to a local repository from Github. We have used liberica-11 as a jdk which can be downloaded from there. Liberica jdk includes open JavaFX. This jdk can be added through the IntelliJ IDEA from the File -> Project Structure -> Project as above.

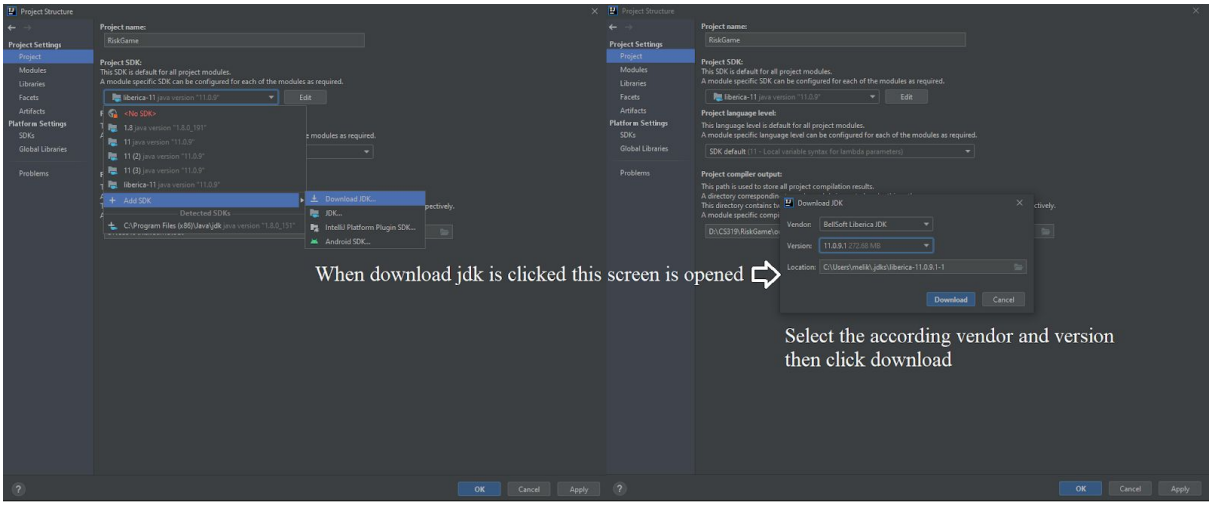

Figure 7 : IntelliJ IDEA Downloading JDK

In order to run game properly display resolution should be 1920 x 1080 and scale should be %100. After setting the correct jdk, you should be able to build and run the Main.java class.

5. Work Allocation

Emin Adem Buran :

- Contributed to reports.
- Implemented most of the user interface designs and controller classes of it.
- Connected the user interface to the game logic.

Onur Oruç :

- Contributed to reports.
- Implemented most of the game logic
- Helped to connect the user interface to the game logic

Ömer Yavuz Öztürk :

- Contributed to reports.
- Implemented most of the game logic
- Helped to connect the user interface to the game logic

Melike Demirci :

- Contributed to reports.
- Implemented the load game user interface and controller
- Write javadoc comments of the user interface classes.

Yusuf Ziya Özgül :

- Contributed to reports.
- Implemented the game logic and load game and save game methods.
- Write javadoc comments for methods in game logic.
- Contributed to final report.