Bilkent University

Department of Computer Engineering

CS319 Term Project

RISK

Final Report

Group 2I

Ahmet Kaan Uğuralp – 21803844

Selcen Kaya – 21801731

Alperen Can – 21601740

Şükrü Can Erçoban – 21601437

Selin Kırmacı – 21802177

Instructor: Eray Tüzün

Teaching Assistant(s): Barış Ardıç, Emre Sülün and Elgun Jabrayilzade

Table of Contents

1. Implementation Process

The implementation process started with creating a shared vision for our project and distribution of tasks among the group members. We decided to divide the tasks according to each member's strong suits in programming, so two members were responsible for the frontend package which includes the user interface components, and three members were responsible for the backend package which includes all game objects and logic. This division was not a strict one as everyone could help others with their task if needed.

Before the first iteration of reports, most classes and some skeletal code and functions were created to help with diagrams and communication across team members. Throughout the following weeks, we continuously added the code needed for each class for the backend package since it's the essence of the project. Afterwards, the frontend package's contents were created and constructed. Then the integration of both the frontend and backend packages were done, with many tests and bug fixes. These were done right before the second iteration of reports, by which some changes and improvements were made regarding the design elements and structure. In the last week, more testing and bug fixing took place and some minor features were added since there was still time.

The changes made to the code and design are explained in the section below. Even though we could not do everything we wanted initially, we added a lot of additional features not mentioned in previous reports to make up for it. The state of the game is finished and working correctly.

2. Design Changes and Improvements

There are some changes regarding the design of the project from the previous reports which are explained below in different subheadings for ease of readability.

2.1 Added Parts

More classes were added to the frontend package, which includes our View and Controller components of MVC design. The new classes helped us with extendibility, maintainability and readability.

Saving and loading game options were added to the project to increase the playability of the game since a game might be longer than anticipated. These features allow the players to leave the game whenever after saving and continue playing where they left.

Some functions and attributes were added to the classes in addition to the ones in class diagrams presented in previous reports to increase readability and maintainability. For example, we added isNeighbor(territory) to neighbors of the chosen territory and we added decreaseImmunityCounts() to decrease immunity count of territories and remove the territories which have no more immunity.

A new restriction was added to the attack numbers. A player can only attack 3 times in one round. This addition was added to limit overpowered players' capabilities in conquering territories and results in a new strategy style of the game.

A new game mode, called the secret mission mode, was added to the game in order to extend the playability of the game. By selecting this mode before the game begins, each player will be assigned a secret mission which accelerates the game since these missions take shorter to achieve than world domination. By this addition we offer 2 different game modes.

There is a mini game that the players can play during the Risk game. This addition adds attractiveness to the game and a new element to the game which can help players gain advantages and not get bored. In the mini game, the player tries to collect troops by left and right arrow keys and the collected troops are given the player's hand to use.

We added a timer for each turn and attack phase in order to limit the waiting times for other players. Since the game is played on a single computer, this feature adds continuity of the game for the players. The maximum time that a single player can have in one turn in 1.5 minutes. This also adds a strategy style alike chess, where the players have to think about their next moves long before playing.

2.2 Removed Parts

The GUIController class which was supposed to be the Controller of the MVC design was removed and the part of View and Controller were merged together in the GUI class. The view component of our MVC design and the controller are gathered under the frontend package. The reasoning with this decision was that the coding and maintaining was more intuitive and smooth.

The Artillery and Calvary classes from the class diagram in the first iteration analysis report were removed since in the Risk board game, they are used to free up and declutter the board scape and we do not have such a concern in our digital implementation. Instead, each player gets the standard amount of soldiers represented with infantry numbers.

The commander properties and advantages feature which was proposed in the previous reports were removed. There are still different commanders all players can choose but they offer no additional benefits since it has the potential to make the balance of power uneven.

2.3 Changes to Diagrams

Based on changes explained above, some diagrams need to be updated. The most important ones, class and subsystem decomposition component diagrams are given below in their new form.

2.3.1 Class Diagram

The class diagram given in the next page includes all changes from the second iteration of the design report.

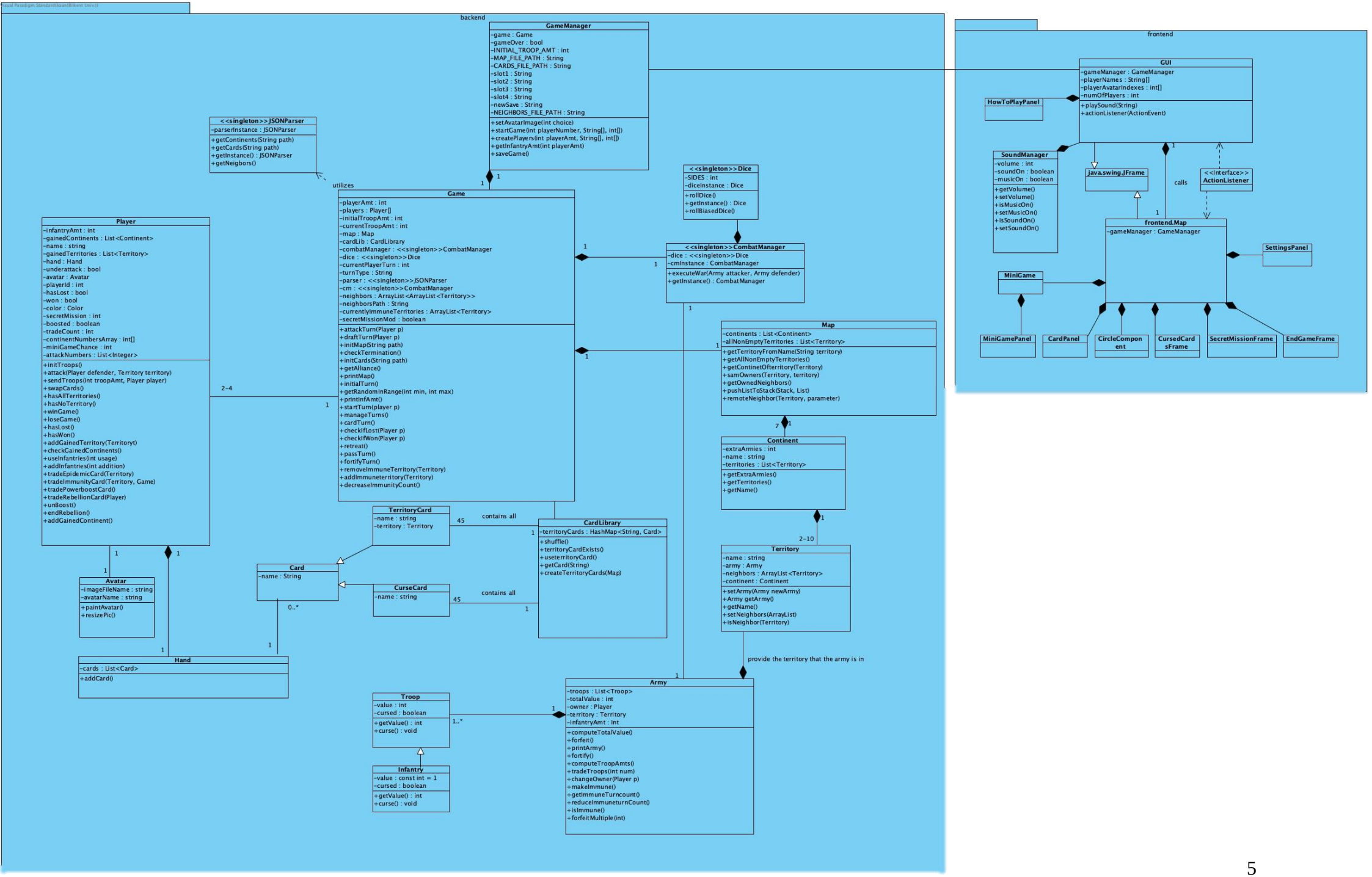

Figure 1: Class Diagram of the Project

2.3.2 Subsystem Decomposition

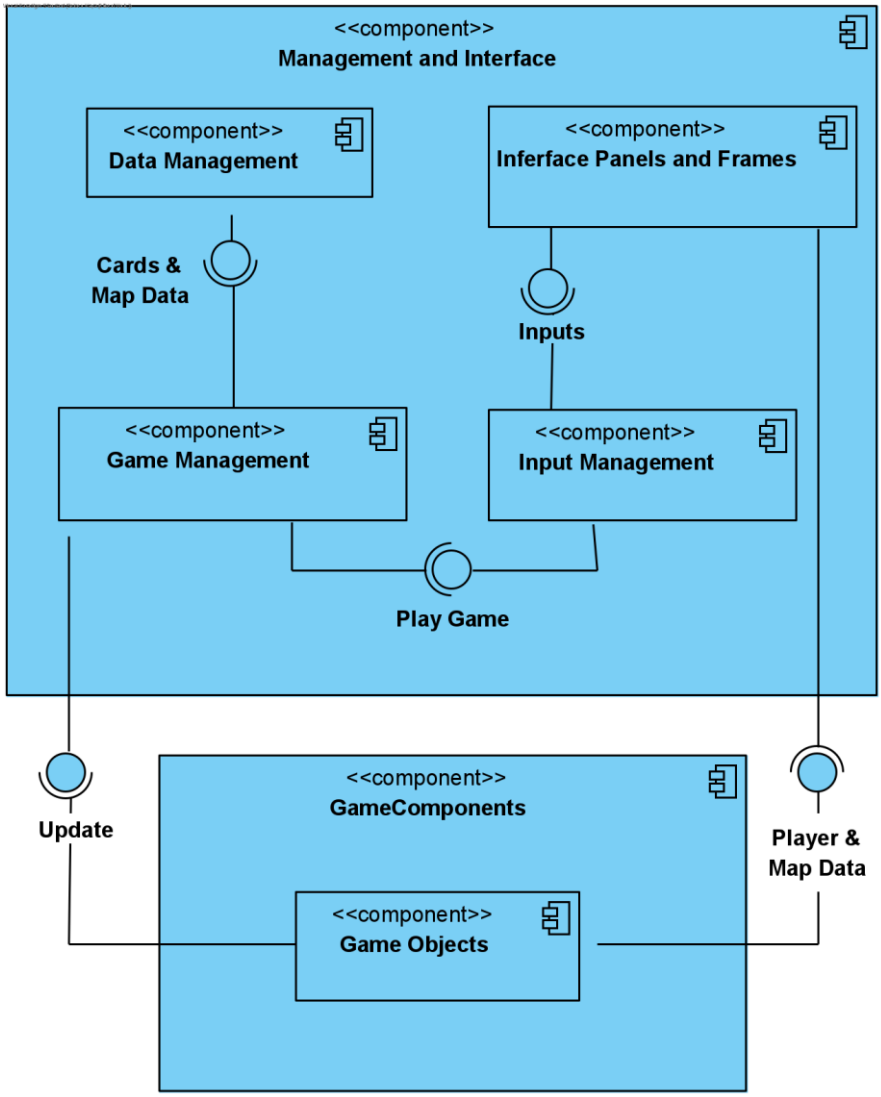

Figure 2: Subsystem Decomposition Component Diagram of the Project

Initially, our subsystem was composed in the style of MVC design, however during the implementation this choice was changed. The view and controller were merged together into one, as mentioned below and this subsystem is named Management and Interface in the diagram above. This includes handling of all user interface components and the game management using the objects in the GameComponents subsystem.

3. Work Allocation

Ahmet Kaan Uğuralp

- Worked mostly on the backend side and contributed to all of the backend classes.
- Worked on testing and debugging the game in both frontend and backend.
- Designed and drew class and object diagrams.
- Contributed to analysis, design and final reports.

Alperen Can

- Worked on frontend side and coded how to play, settings and load game panels.
- Implemented start game menu which includes avatar and personal feature selections.
- Added sound effects.
- Designed and drew use case models.
- Contributed to analysis, design and final reports.

Şükrü Can Erçoban

- Worked on backend side and coded combatManager class and attack methods.
- Worked on testing the game and designed the game trailer.
- Designed and drew activity diagram.
- Contributed to analysis, design and final reports.

Selin Kırmacı

- Worked on the frontend side regarding user interface and coding of the majority of the game.
- Worked on testing and debugging the game in both frontend and backend.
- Managed the group, work allocation and group meetings.
- Coded and designed the mini game.
- Custom designed territory, cursed and secret mission cards.
- Added music to the game.
- Contributed to analysis and design reports.

Selcen Kaya

- Worked on the backend side and coded JSONParser class and related documents.
- Worked on the frontend by creating all territory cards.
- Designed and drew sequence diagrams, state diagram and subsystem composition.
- Contributed to analysis, design and final reports.
- Created the slides for both presentations.

4. User's Guide

This section includes information about the system requirements for running the game, instruction on how to run the game as well as how to navigate and play our implementation of Risk.

4.1 System Requirements

Risk game requires Java SDK and Java Virtual Machine to run. Any computer which has Java is able to install and run the game.

The build instructions below require you to have Java 1.8 version (can be downloaded from https://www.oracle.com/tr/java/technologies/javase/javase-jdk8-downloads.html) as well as IntelliJ IDEA (can be downloaded from https://www.jetbrains.com/idea/download) installed.

4.2 Build Instructions

Preparations:

- Install these or make sure that these are installed in your computer: Java 1.8 version (https://www.oracle.com/tr/java/technologies/javase/javase-jdk8-downloads.html) and IntelliJ IDEA (https://www.jetbrains.com/idea/download).
- Download the project or make sure that the project is downloaded from https://github.com/selinkirmaci/cs319-risk-2I. The project is private, so if you need access please contact us.

Build Instructions:

- Open IntelliJ and click open new project button. Find the project file you downloaded on your computer.
- Open pom.xml (located in backend) file as project.
- Open File > Project Structure. Set Project SDK to 1.8 and set Project language level to 8 – Lambdas, type annotations etc. Click Apply and Ok.
- Build the project through Build > Build Project. Find GUI class located in backend > src > main > java > frontend in the left side of the screen, project view.
- Right click the GUI class and click on Run 'GUI.main()' option.
- Enjoy the game. ☺

4.3 How to Play

Game guide will be explained in eight subsections for clarity.

4.3.1 Main Menu

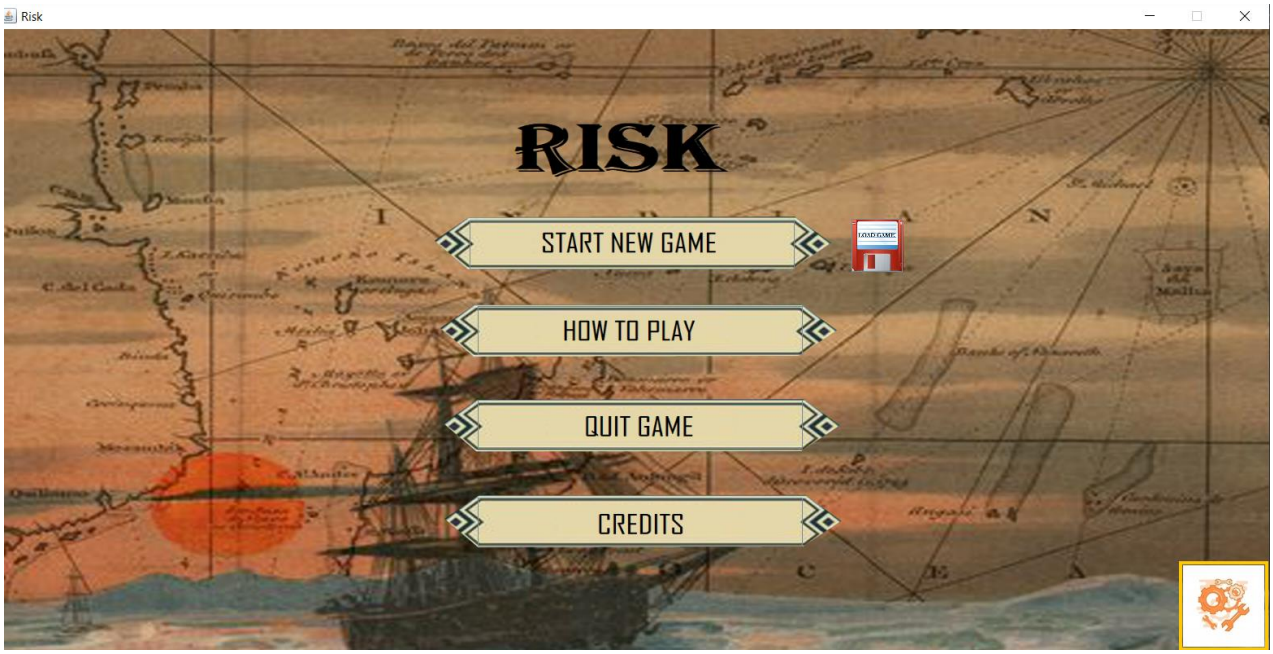

Figure 3: Main Menu

Player can access start new game menu, load game menu, how to play screen, credits and settings from main menu.

4.3.2 How to Play Menu

Figure 4: How to Play Menu

Information about game modes, map, cards and more can be found at this panel. How to play menu consists of several pages. User can view next or previous pages by clicking the buttons at the bottom of the screen.

4.3.3 Settings

Figure 5: Settings Menu

Settings button is located at the bottom-right of the main menu. Music volume of the game can be changed by using the slider. In addition, Risk game includes sound effects for buttons and for some cases such as winning the game. All sound effects can be enabled or disabled by altering the status of the checkbox. Before quit settings, player must click save button to save the latest changes.

4.3.4 Credits

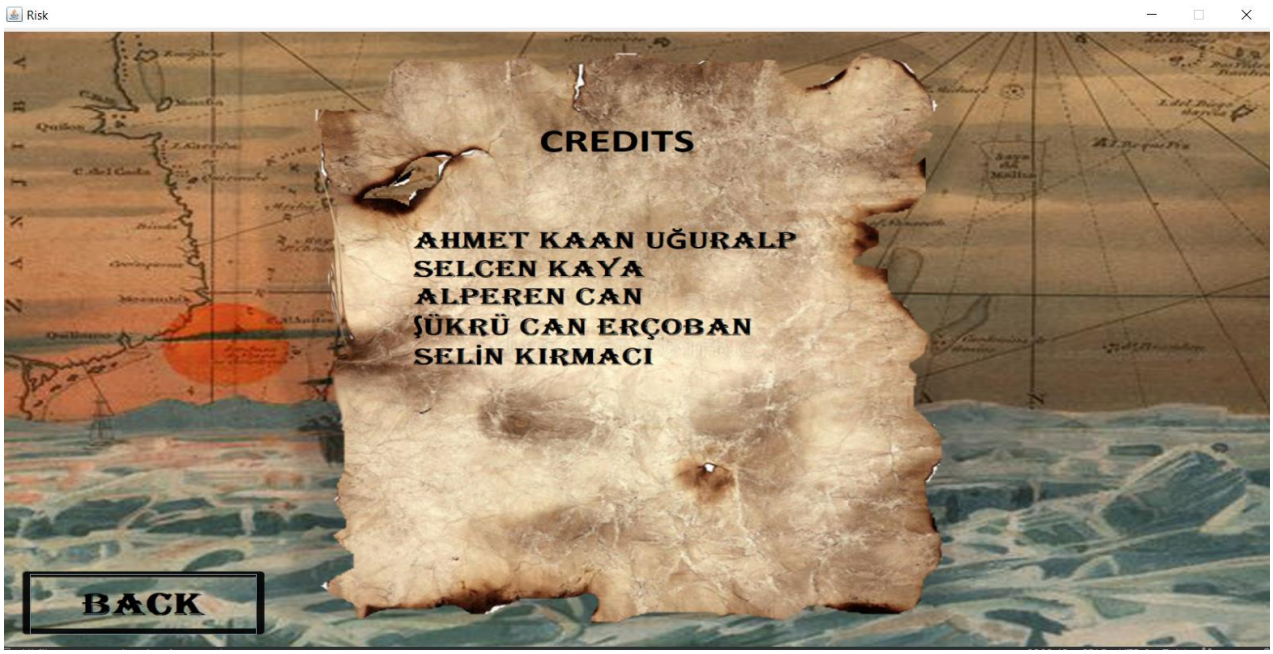

Figure 6: Credits Screen

Names of the developers can be seen in this panel.

4.3.5 Load Game

Figure 7: Load Game Menu

Player can access load game menu from main menu by clicking the floppy disk icon next to the start new game button. Previously saved games can be loaded from this menu. Saved date and time of the games can be seen on the non-empty slots. Clicking the delete button will clean the slot next to it.

4.3.6 Start New Game

Figure 8: Start New Game Menu

Players can choose colours, avatars and nicknames from this menu in order to represent themselves. These features must be unique to start the game, otherwise a warning message will be displayed. Nicknames should be entered to textboxes. Colours and avatars can be chosen by clicking select colour and select avatar buttons. From the top-right of the screen, number of players can be determined. Moreover, enabling the secret mission checkbox will set an alternative game mode. Game can be started by clicking continue button.

4.3.7 Gameplay

Figure 9: Main Gameplay Screen

The avatar of player with turn will be highlighted by the color he has selected. The number of extra soldiers in a hand of a player can be seen below the avatars. Color of the number of troops on a territory indicates the owner of that territory. Players can pass their turn by clicking next player button. In addition, expiration of time will also result in passing turn. Players can see the remaining time of their turn at the top of the map, next to the pause button.

Figure 10: Draft Pop-up Screen

Players can locate their extra soldiers to their territories by clicking draft button. Number of extra troops will be reduced from players' hand after the drafting process.

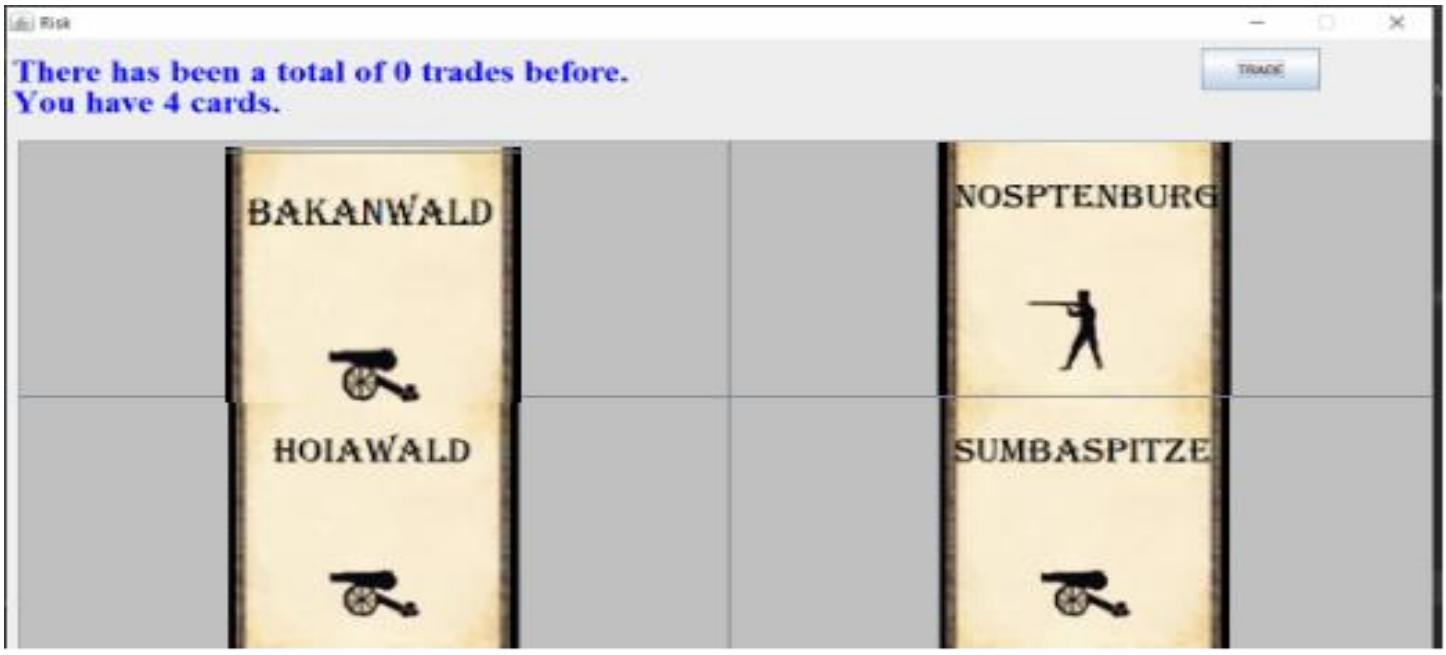

Figure 11: Owned Cards Information Screen

The number and type of owned territory cards can be viewed by clicking card info button. Cards can be traded with extra troops from this pop-up screen. Moreover, number of trades done by players is shown at the top-left of the screen.

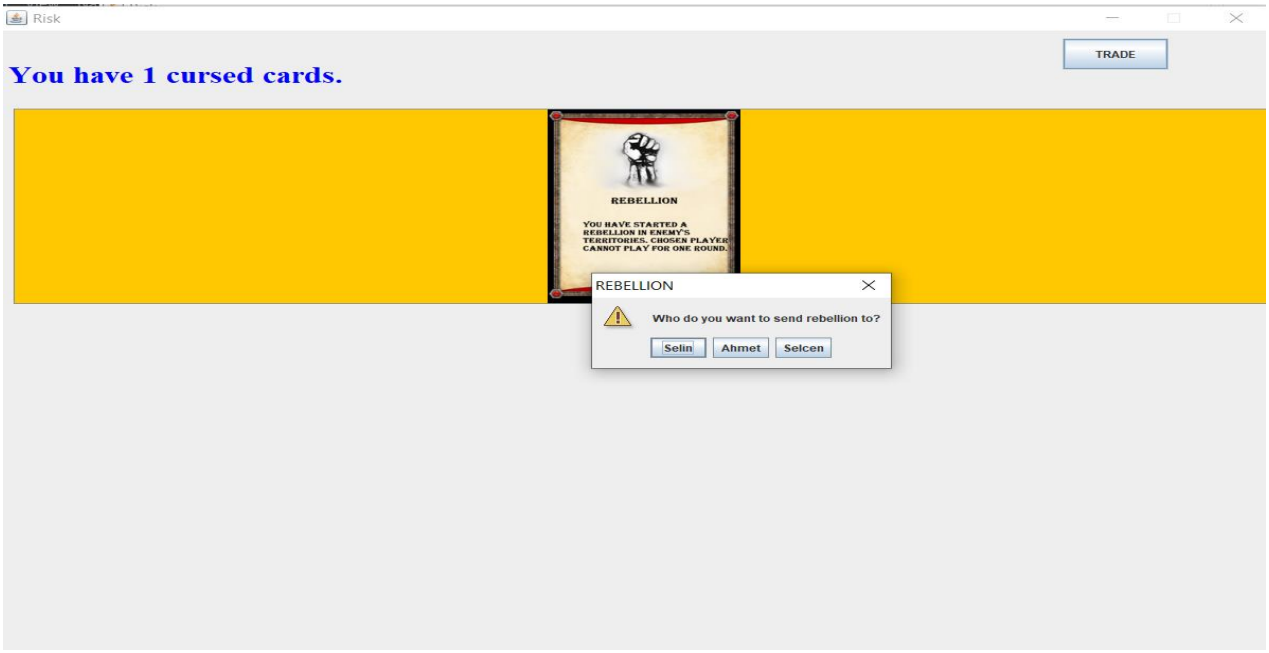

Figure 12: Owned Cursed Cards Information Screen

Curse cards can be viewed by clicking curse cards button.

Figure 13: Secret Mission Pop-up Screen

If secret missions were included by enabling the checkbox from start game menu, secret mission button will set to visible. Players can view their secret missions by clicking it.

Figure 14: Attack Screen

In order to attack a territory, player with turn must click one of his owned territories. This will set the territory where he will attack from. Following, an enemy territory should be

clicked to attack to. Ultimately, player should click attack button. The aggressor and defender players may change the number of dice to be thrown from attack screen. Clicking roll dice button will start the clash. Attacker cannot stop the attack once it has begun. During the war, the defender can offer alliance to another player by clicking alliance button. If the offer is accepted, ally player will enter the number of troops that he wants to send to support defender's territory.

The way using fortify and retreat functions is similar to using attack button. Initially, the source territory should be selected. Secondly, target territory will be determined. Following, player can click the button to fortify or abandon a territory, depending on the button clicked.

4.3.8 Pause Menu

Figure 15: Pause Menu Pop-up

The game can be paused by clicking pause button at the top of the screen. Players can save the game or access settings and how to play menus from pause screen.