Bilkent University

Department of Computer Engineering

CS 319

Object Oriented Software Engineering

*Game of “Risk” Final Report*

Group 1-E

Mehmet Ege Acıcan-21602186

Ekin Üstündağ-21602770

Mehmet Akif Kılıç-21502487

Seman Arda Yuvarlak-21501402

Contents

[1 Introduction 1](#_Toc254263637)

[2 Design Changes 1](#_Toc254263638)

[3 Lessons Learned 1](#_Toc254263639)

[4 User’s Guide 1](#_Toc254263650)

4.1 System Requirements and Installation\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

4.2 How to Use\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

4.2.1 Game Overview\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.2 Menu Operations\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.3 Starting a New Game\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.4 Loading a Previous Game\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.5 Adjusting the General Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.6 Highlighting the Statistics\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.2.7 Adjusting the Game in Pre-Game-Settings Menu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.3 Gameplay Options\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Work Allocation\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.1 Mehmet Ege Acıcan-2160218\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.2 Mehmet Akif Kılıç-21502487\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.3 Seman Arda Yuvarlak-21501402\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.4 Ekin Üstündağ-21602770\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.5 Ali Ata Kılıçlı-2130\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.Introduction

In this report, our aim is to discuss the implementation process, the design decisions, design changes and offer an users guide on Game of “Risk”. This section specifically focuses on the implentation process and the current status of the implementation.

After redesigning our game design, we have made specific updates to the Game of “Risk”. In the design report, we have divided our main system into three subsystems. This allowed us to distribute the work*.*

Our meetings were face to face as much as possible. If face to face meeting was not possible, the meeting was done by whatsapp. In these meetings, we discussed the design, the structure and took specific design choices along with updating the implementation.

After the implementation of the menu interactions and main gameplay, our focus shifted on fixing the errors of the main game, then it shifted towards saving and loading different games and finally it shifted towards to adding extra features to the game. Although we have successfully included in-game additions to the game such as extra troop types and a different battle logic. The game now has 5 troop types that are Soldier, Cavalier, Cannon, Plane and Tank. Also, battle logic works in a different way. Firstly, the Attacking terrirory must have higher troop number than defending territory. Then, from the attacking territory, n number of troops are chosen randomy where n is the number of troops in the defending territory. Finally, randomly each attacking troop pairs with a defending troop. The winners and losers of the individual troop battle are determibed by their attack points and defence points. However, additional gameplay mods such as time mod and Golden Territory mod could not be added. Also, saving and loading of the game could not be added.

In GUI Subsystem; main menu, pre-game settings menu, InGameGameplay menu, credits menu, display help menu and adjust settings menu were inplemented. Also, their connection with each other and transitions between the menus were handled.

In Game Logic Subsystem; Specific classes that we had were updated and redesigned. These classes include: Troop, Territory, Game and Player. Specific enumerations were added. Also, for the textures and color changes, texture class and imageProcessor class were added.

Data Subsystem were not implemented.

User interfaces were handled by using fxml files. Buttons, sliders and other parts of the user interfaces were integrated into these fxml files for increasing the simplicity.

2.Design Changes

Continuing from our iteration 1 implementation process, we have firstly started to fix our problems and tried to offer solutions to the parts where our game had problems. We have kept the MVC model and we have improved by adding more model classes, certain game screens and controller classes. Each menu has their own controller in order to increase independence of the menu screens and thus allowing sudden changes to the game. This was helpful for our Adjust settings menu where you are able to reach while playing the game. This menu was supposed to be a part of the Pre-Game-Settings menu however we have decided to just add a separate menu for gameplay mods.

Firstly, we have focused on the GUI layer where our project was lacking during the first iteration. We have started to create the screens by creating fxml files using scenebuilder then we have allowed the transitions between these menus by adding a specific controller for each of the menus and then allowing a transition function for specific parts of the screen such as buttons in these menus. Therefore, we have made the user interface menus and their connectinos with each other.

Secondly, we have improved our game logic and made the gameplay mechanics to function. We have reconsidered our battle logic and redesigned it. We have added new features such as new troop types to the game along with their new stats based on our new battle logic. Also, we have added texture and image processor classes in order to adjust the color changes that are happening in the game.

Finally, we have made a connection between these two layers and thus we have allowed the main gameplay of the game.

To sum up, we have allowed the players to play the Game of “Risk” in its classical mod with its in game extended features.

We were mainly focused on the GUI and Game Logic layer. We could not have made progress in the Data Layer although we have preferred to use Gson files as our way to save the files.

3.Lessons Learned

During our implementation of the game, we have understood the importance of analysis and design stages and most importantly its synchronous action with the implementation. We have tried to approach to the project in a Waterfall lifecycle approach which is focusing on speculative requirements and design processes before implementation and implementing the project based on the speculated requirements and designs. However, we have learned that iterative approach would be better for this project, that is, implementing and completing the requirements and design synchronously. We have learned that implementation can allow us to fix the errors that we could not foresee in the design and therefore allowing to make our design better.

Another valuable lesson that we have learned is the importance of groupwork and the importance of work distribution. In our project, we have separated our tasks and we tried to implement the loose coupling structure to our roles and works as well however we have realized that we have limited independency since our tasks are supposed to work cooperatively and therefore we had to be in contact with each other about our projects regularly. Also, we had understood that work distribution should be equal and therefore we should not rely on one person too heavily while not sharing the work with other members.

Other lessons that we have learned is that flexibility can be helpful for making progresses in group work. Also, we have witnessed the effect of change in a software project. We had to rethink about our design for multiple times and as we progress in the project, we are certain that we will reconsider specific updates on our project again

These were the lessons that we have gathered in the first iteration. In the second iteration, we have once again seen the importance of working together and we have understood that the projects can not be taken single handedly. All of the project members should be participating in order to fully complete the product.

Also, we had understood the importance of using the time efficiently so that the productivity is maximum.

4. Users Guide

4.1 System Requirements and Installation

Game of “Risk” can be runned in all platforms and does not need any installation. It is a single executable JAR. However, in order to be able to run the JAR, computer should have Java Runtime Environment installed.

4.2 How to Use

4.2.1 Game Overview

Game of “Risk” is a strategic conquest game which is designed for desktops and laptops. After the initialization, the user sees the main menu that allows the player to start a new game, load a previous game, adjust the settings, see the credits or exit the game. Players are able to play in 3 different gameplay mods and they are able to save their game, load their saved game and delete their previous games.

4.2.2 Menu Operations

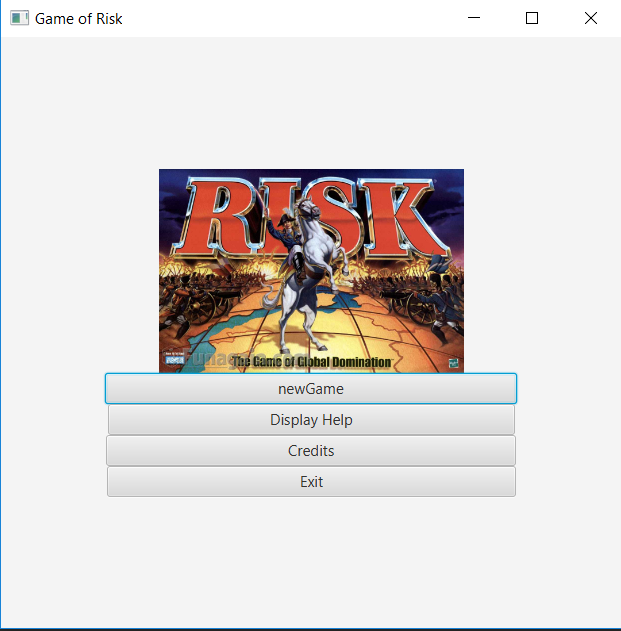


Figure 1: Main menu of the Game of "Risk"

4.2.3 Starting a new game

Players are able to start a new game by selecting new game from the main menu, then they choose how many players will play the game by choosing from the slider. Then they will enterb the names of these players and then they will start the game. The number of player selection and name entering is done in the Pre-Game Settings Menu.

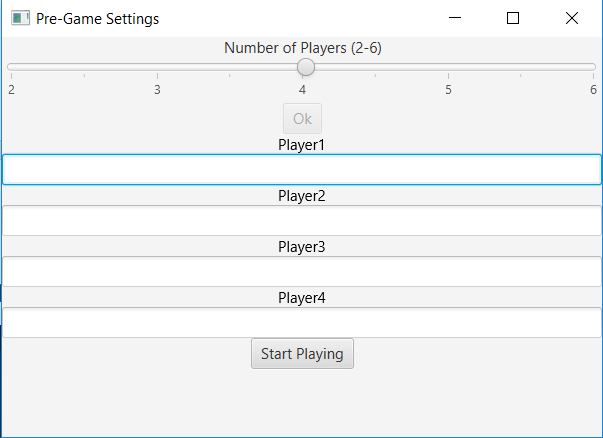
******

Figure 2:Pre-Game Settings Menu

4.2.4 Loading a previous game

Players should be able to load a previously saved game from the load game. However, the implementation process is still going on.

4.2.5 Adjusting the general settings

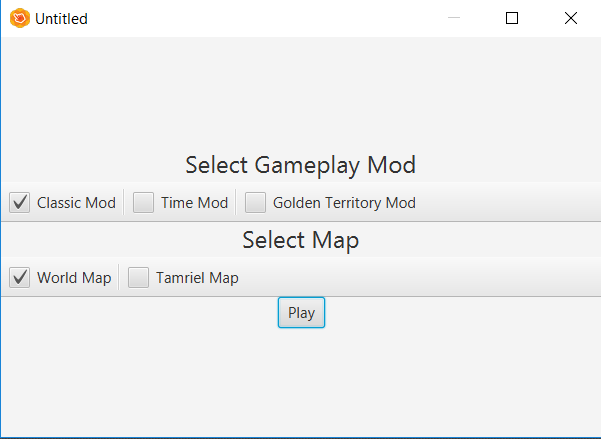


Figure 3: Adjust Settings Menu

The players are able to adjust settings in the adjust settings menu in the gameplay menu. From this menu, when they enter to the Adjust Settings Menu: They are able to change the gameplay mod of the game and they should be able to adjust the time-golden territories of the game depending on the gameplay mod.

4.2.6 Highlighting the Credits and Display Help

The players are able to view credits and display help from the main menu and they are able to se the people who have made the game from the credits menu and they are able to get specific information about the game from Display Help menu.

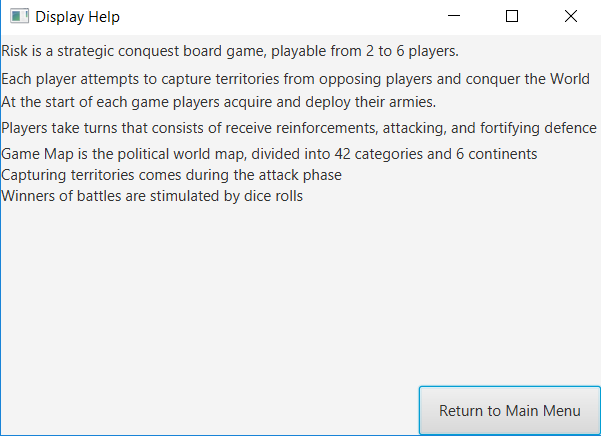
******

Figure 4: Display Help Menu

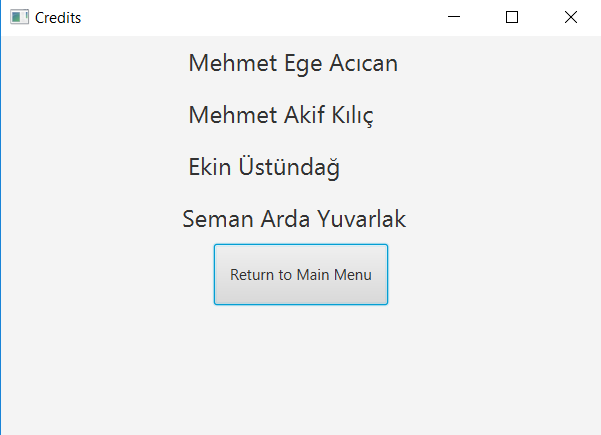


Figure 5: Credits Menu

4.2.7 InGame Gameplay

The players are adjusted a random color at the beginning of the game and then they receive specific number of unit right at the beginning. Then, based on the cost of the troops, the players start to place troops to their territories. Then, once the first turns are over and then all the troop rights are used, the players can attack to their opponents territories. The game ends based on the specific conditions of the game mod. If the player plays on the classical mod, the player must conquer every territory in order to win. If the player plays on the golden territory mod, the player must conquer golden territories. If the player plays on the time territry, the player must win as much territory as possible before the clock runs out.

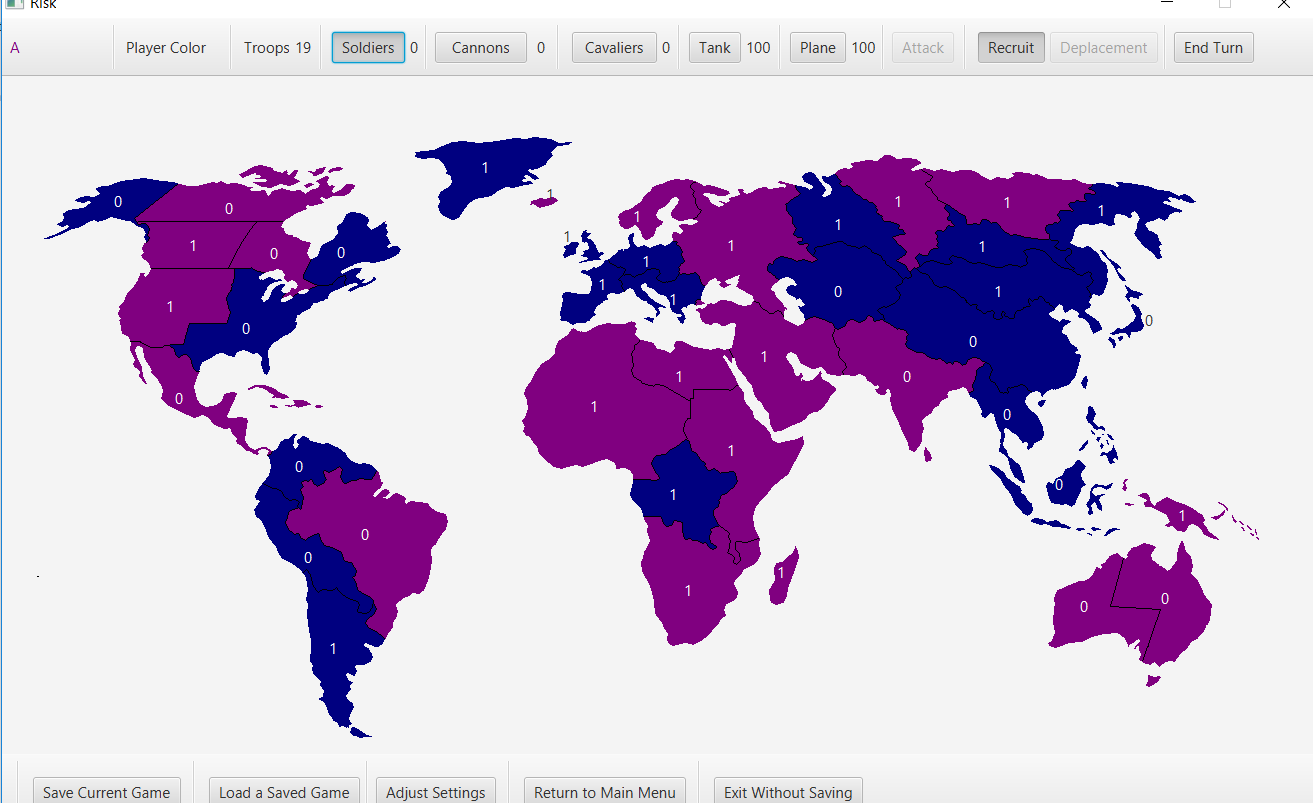


Figure 6: InGame Gameplay of Game of "Risk"

5.Work Allocation

The following shows which group member has done which part of the project in the first and second iterations.

Mehmet Ege Acıcan

Iteration 1

Analysis Report

1. Responsible for writing the introduction part of the Analysis Report
2. Responsible for writing the Game Overview part of the Analysis Report
3. Responsible for writing the Functional Requirements part of the Analysis Report
4. Responsible for writing the Non-Functional Requirements part of the Analysis Report
5. Responsible for designing the Use Case Diagram and writing its use cases descriptions
6. Responsible for designing the State Diagrams and writing the diagrams descriptions
7. Responsible for designing the Sequence Diagrams and writing the diagrams descriptions.
8. Responsible for the descriptions of the User Interface part of the Analysis Report.

Design Report

1.Responsible for writing introduction part of the Design Report

2. Responsible for writing the High Level Software Architecture part of the Design Report.

3. Responsible for writing the GUI layer part of the Subsystem Services part of the Design Report.

4.Responsible for the low level design part of the Design Report

Final Report

1. Fuly responsible for the Final Report in Iteration 1

Presentation

1. Responsible for making the full presentation in powerpoint.
2. Responisble for presenting the Functional Model part in the presentation
3. Responsible for presenting the Dynamic Model part in the presentation.
4. Responsible for presenting the High level Design part in the presentation.
5. Responsible for presenting the Low level Design part in the presentation.

Implementation

Iteration 2

Analysis Report

1. Responsible for editing and writing Functional Requirements part
2. Responsible for editing and writing the Non-Functional Requirements part
3. Responsible for editing and redesigning the Use Case Diagram and writing its use cases descriptions.
4. Responsible for editing and redesigning the State Diagrams and writing its descriptions
5. Responsible for editing and redesigning the Activity Diagrams and writing its descriptions.
6. Responsible for editing and redesigning the Sequence Diagrams and writing its descriptions.
7. Responsible for editing and redesigning the Class Diagram and writing its descriptions.

Design Report

1. Responsible for writing the High Level Software Architecture part of the Design Report and redesigning the Deployment Diagram and writing its descriptions.
2. Responsible for writing the Subsystem Services part of the Design Report along with describing its respective diagrams.
3. Responsible for writing the Low level design part of the Design Report.

Final Report

1. Responsible for the Introduction part of the final Report
2. Responsible for the Design Changes part of the Final Report
3. Responsible for the Lessons learned part of the Final Report
4. Responsible for the Work Allocation part of the Final Report

Presentation

( Presentation slides have not been edited yet)

Implementation

1.Responsible for designing the fxml files in the scenebuilder for the GUI of the game.

2.Responsible for editing the model classes and connecting the model classes with the GUI of the game.

Mehmet Akif Kılıç

Iteration 1

Analysis Report

1. Responsible for writing the class diagram and its descriptions of the Analysis Report.

Design Report

1. Responsible for writing the data layer part of Subsystem Services part of the Design Report.

Final Report

Presentation

1. Responsible for presenting the class diagram of the presentation

Implementation

1. Responsible for writing the model classes.

Iteration 2

Analysis Report

Design Report

1. Responsible for editing and rewriting the Inroduction part of the Design Report.

Final Report

1.Responsible for writing the User’s Guide part of the Final Report.

Presentation

( Presentation slides have not been edited yet)

Implementation

1. Responsible for editing the model classes.

Ekin Üstündağ

Iteration 1

Analysis Report

1. Responsible for Activity Diagrams and their descriptions

Design Report

1. Responsible for Gameplay Layer and its descriptions

Final Report

Presentation

1. Responsible for Video
2. Responsible for the presentation of the non-functional requirements in the presentation

Implementation

Iteration 2

Analysis Report

Design Report

Final Report

Presentation

( Presentation slides have not been edited yet)

Implementation

Seman Arda Yuvarlak

Iteration 1

Analysis Report

Design Report

Final Report

Presentation

1. Responsible for presenting the introduction part of the presentation

Implementation

1. Responsible for editing the model classes

Iteration 2

Analysis Report

Design Report

1. Responsible for describing the Game class that belongs in the Game Object Manager subsystem of the Game Logic layer

Final Report

Presentation

( Presentation slides have not been edited yet)

Implementation

Ali Ata Kılıçlı

Iteration 1

Analysis Report

1. Responsible for designing the UI in the User Interface part in the report

Design Report

Final Report

Presentation

Implementation

1. Responsible for designing the UI part of the game( Did not do it).

Iteration 2

Analysis Report

Design Report

Final Report

Presentation

( Presentation slides have not been edited yet)

Implementation

Notes: Ali Ata Kılıçlı has withdrawn from the course.

Notes: There is no writing mistake in the Work Allocation part of the report. These reports show exactly which group member has done which part of the project and it shows how much contribution that they have to the project.