CS 319 Term Project Final Report

Alara Yaman 21101164
Berfu Deniz Kara 21201662
Can Demirel 21401521
Muhammed Emin Aydın 21002035
Servan Tanaman 21201977

Supervisors: Gülden Olgun, Eray Tüzün

Project Group Name: Oldies but Goldies

Project Topic: Risk Board Game

1. Introduction

In this project, we design the box game RISK to digital version by implementing it using Java. We choose Java that will help us implement our code in an easier and quicker way. We preferred Apache Netbeans as development environment because of its UI features. We are using a MySQL database to hold our players and game data. Rather than the standard version, we add new innovative features to our game. These features make our game more interesting. For example, we add epidemic which reduces the unit count or we add seasons to our game which also changes the unit count. Moreover, we add new content which has 3 territories in the Atlantic Ocean. All of these new features make our game more enjoyable than the standard version. After writing the code we played it and make sure it works, we pushed it to our GitHub page.

2. Design Changes

2.1 Software Changes

- New getter and setter methods added for dynamic UI interaction
- Quests defined as different objects
- Exceptions and boundary checks added
- Seasons Added that reduce attackers highest dice value by 1.
- Epidemics Added that decreases the recruit amount at %5 chance
- New Continent that contains 3 territories added

2.2 GUI Changes

- Background images added to every territory.
- Buttons are redesigned.
- Winter Effect Added
- Epidemic Effect Added
- New Continent added to Game Board

3. Lessons Learnt

During this design process, we learnt to use our time better and wiser because we saw that there is a heavy workload in the mean of both coding and report writing. At first we were starting to write the reports 3-4 days before and was finishing just before the due date. But for the second iteration process, we chose to start 1 week early.

Another early problem was that it was not easy to arrange a meeting time because all the group members are taking different different courses so there is no available time during day time but we manage to overcome this by renouncing our free evening time.

Other than that we learnt from the iteration 1 that coding UI and game engine separately is caused a huge amount of time loss so for the iteration 2, we implemented our codes together in order to use our time more efficiently.

As a group we learned that having divergent expertees is a helper to design and implement a game because a project design requires more than just coding, report writing, high-level and low-level designs and artistic perspectives are also needed.

We also learned that an error which occurs during design process can cause time loss because implementation will be errored too. So low-level design and high-level design are very important and require time to think on it.

It was a challenge to study as a group when nobody was really close with each other at the beginning but even though with some minor flows, we manage to study together as a group and spent time and effort for this project. We can say that we learnt more about how to be a group and get through a project process.

4. User's Guide

4.1 System requirements & installation

Minimum system requirements:
-256 MB RAM
-100-120 MB storage space
-Windows (XP and higher) OS
-Sound card

The game can be executed directly from RISK.exe. There is no need for a setup process. JAVA should be installed to the system that the game will run.

4.2 How To Use

4.2.1 Beginning

4.2.1.1 Login

In our game for the login system, we use username and password. If wrong username or password we get an error. We want every single player to login the system because we want to keep their informations in our database. If they quit the game, they can restore it.

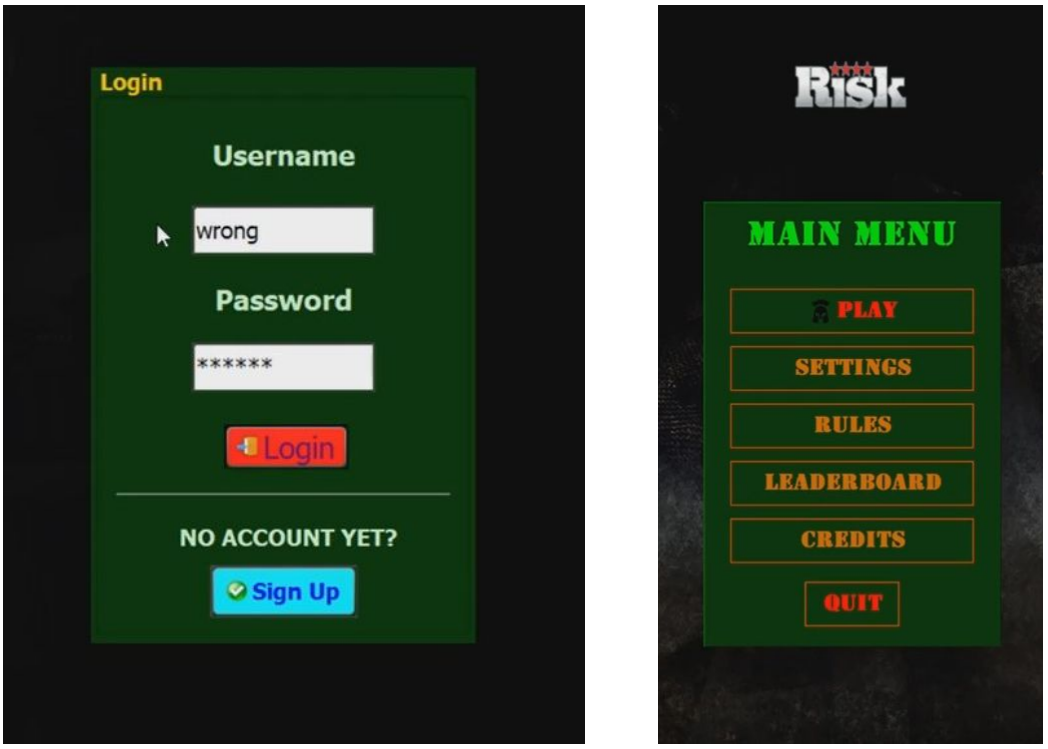

Figure 1: Login and Main Menu Frame

4.2.1.2 Main Menu

In our game, after login the system the main menu opens. In the main menu, there are several buttons. Play button for starting the game. Settings button for on/off the music and theme selection. These two features make the game more enjoyable. Rules button for reading the game manual which helps people to understand the game easily. Rather than standard version our game has different rules. Also, if the person did not play the game he or she can read the manual and begin to pay. The leaderboard is for showing the game players and score. Credits button and quit button to quit the game.

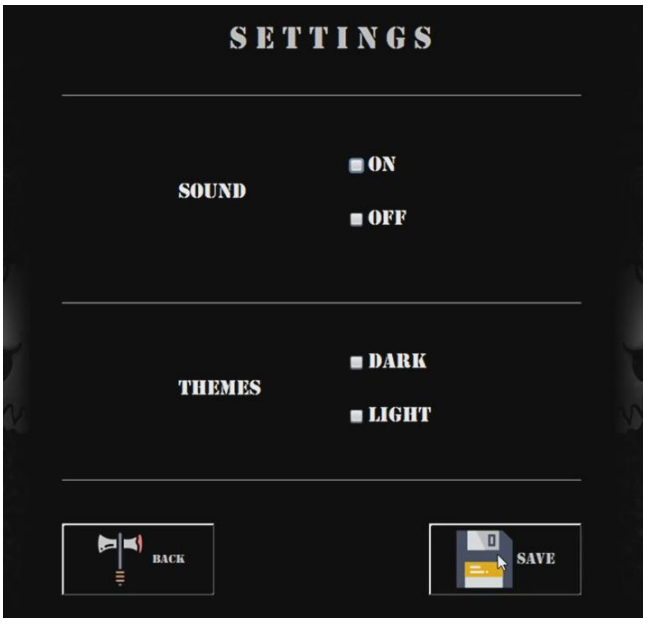

Figure 2: Settings Frame

4.2.2 War Preparation

Every player adds troops to territory in turn at the beginning.

First, choose a territory to enter the preparation screen. If the territory is unclaimed, decide troop amount with "-/+" buttons and click recruit button to claim territory.

Figure 3: Preparation Screen - Recruit Frame

Also, at the beginning of every turn, the player gets an extra amount of troops. These troops can be distributed the same way among to the player's territories.

4.2.3 Attack and Defend the Territory

To attack, choose an origin territory. This territory should be owned by the attacker and neighbor to destination territory. A player can start an attack if origin territory has more than 1 troops.

Figure 4 : World Map

After choosing the territory, war screen will open. In this page first, enter troop amount. (Regardless of the troop amount, there can be a maximum of 3 troops in every fight.) Then click to attack button. According to the result of the fight, the player can continue to attack if there are enough troop or the player can end the fight by click retrieve button.

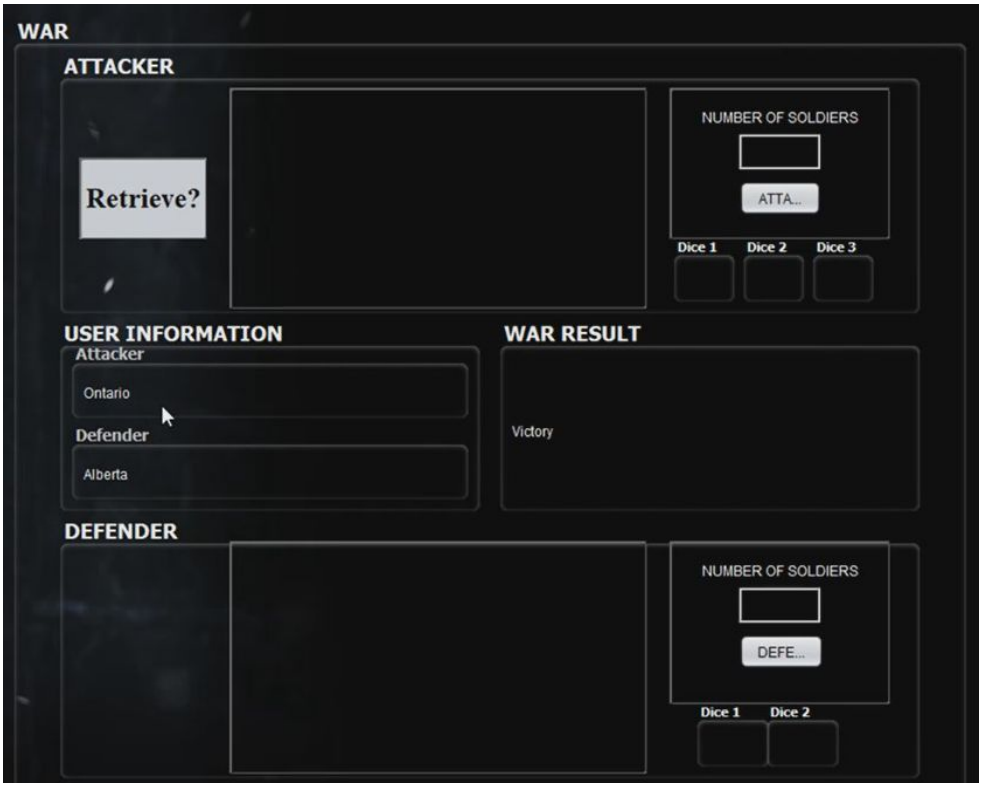

Figure 5: Attack / Defend Frame

To defend, firstly, troop amount should be entered to "number of soldiers" box. The player can defend with minimum 1, maximum 2 troops.

At the end of the fights, if there are no troops left on the defender side, the attacker wins. If the attack side doesn't have any troops to attack, the defender wins.

4.2.4 End Turn

Before the end turn, the player can rearrange troops among owned territories. To move troops, firstly, chose origin territory. In the next page enter troop amount that will be moved. Then chose one of the territories that the player owned as destination territory and click "move" button. The player must leave at least 1 troop on the origin territory.

After the rearrangement, the player can end the turn by click "End Turn" button. Thus turn will pass to the next player.

By the help of the infoboxes located at the top left corner of the war map, players can keep count of the rounds and which player's turn it is.

5.Work Allocation

Alara Yaman

Analysis Report:

- Introduction,
- Overview - In this part i give detail information about the game, playing pieces, game board, rules, cards, how to claim or attack a territory, what are the innovative features we add.
- Sequence Diagrams
- Improvement Summary
- Use Case Model Diagrams
- Overall Report Figure Control
- References Check

Design Report:

- Introduction - Purpose of the System, Design Goals
- Improvement and Summary
- References Check

Final Report:

- Introduction
- How to Use It

❖ Brainstorming for finding innovative features for the game

Berfu Deniz Kara

Analysis Report:

- Use Case Model Diagrams and Their Explanations
- Functional Requirements
- Overall Report Design and Error Check

Design Report:

- High-Level Software Architecture
    - Hardware/Software Mapping
    - Persistent Data Management
    - Access Control and Security

        - Boundary Conditions
    - Overall Report Design and Error Check

Final Report:

- Lessons Learnt

Implementation:

+Debugging on :

- Game Engine Class and Methods
- Game Manager Class
- Quest Objects
- UI and Game Engine integration

❖ Finding additional features for the gameplay

Can Demirel

Analysis Report:

- Use Case Model Diagrams
- Sequence Diagrams
- Activity Diagrams
- State Diagrams
- Class Diagrams

Design Report:

- Game Engine Class Diagram

Implementation:

- Game Engine Class and Methods
- Land Objects
- Player Objects
- Game Manager Class
- Quest Objects

Muhammed Emin Aydın

Analysis Report:

- Non-functional requirements

Design Report:

- low-level design - packages, class interface

Final Report:

- How to use it

Graphic elements (trailer, background, buttons)

❖ Sounds (effect sounds, music tracks)

Servan Tanaman

Analysis Report:

- User Interface
- Mockups

Design Report:

- User Interface Class Diagrams

Implementation:

- Game Database
- Game UI
- Database Connections
- Button Events