CS 319 Term Project Analysis 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 **4**

2. Overview **5**

2.1 General Information about Game 5
2.2 Armies 5
2.3 Game Board 5
2.4 Playing Pieces 6
2.5 Dices 6
2.6 Cards 7
2.7 Object Of the Game 7
2.8 Claiming Territories 7
2.9 Reinforcing Territories 8
2.10 Attacking and Defending 8
2.11 End Turn 8
2.12 Game Modes 8
2.13 Leaderboard 8
2.14 End of the Game 8

3. Functional Requirements **9**

3.1 Additional Requirements 9

4. Non-Functional Requirements **10**

4.1 Usability 10
4.2 Reliability 10
4.3 Performance 10
4.4 Installability 11
4.5 Additional Requirements 11

5. System Models **11**

5.1 Use Case Models 11
5.1.1 Menu-Use Case 11
5.1.2 Create New Game-Use Case Model 15
5.1.3 Play Game Use Case Model 18
5.2 Dynamic Models 20
5.2.1 Sequence Diagrams 20
5.2.2 Activity Diagram 22
5.2.3 State Diagrams 24
5.2.4 Class Diagrams 26
5.3 User Interface 27
5.3.1 Screen Mock-Ups 27
5.3.1.1 Login 27
5.3.1.2 Main Menu 28
5.3.1.3 Tutorial 28
5.3.1.4 New Game 29

5.3.1.5 War 29
5.3.1.6 Settings 30
5.3.1.7 Number Of Players 30
6. Improvement summary 31
7. References 32

1.Introduction

The game we decided to design and implement as a group is called RISK. RISK is one of the famous military strategy board games which is designed for three to six players. We design this board game to computer game using object-oriented language and we choose Java.

The standard version of this game consists of a world map which is divided into six continents: North America, South America, Europe, Africa, Asia and Australia which consists of a total forty-two territories. All of the continents are represented in different colours.

In order to make the game more interesting and competitive for the players, we will add some features to the game. The players will be able to change the background theme of the game if they want to. Also, we will add some background sounds matching with the present acts of players to make our game more sophisticated. Moreover, in our design, we have a tutorial part which helps players to understand the game more easily. If the players face a challenge, they can read the manual. Lastly, we will add a leaderboard to show the scores of all the players. Players can look at the leaderboard and arrange their strategy according to this table, in order to be the winner of this competitive game. Moreover, we add new mission cards which have different missions. These new missions make the game more interesting. Also, we add a new continent in the Atlantic ocean which has 3 territories. This land is unique for our game. With this new feature, we have 45 territories.

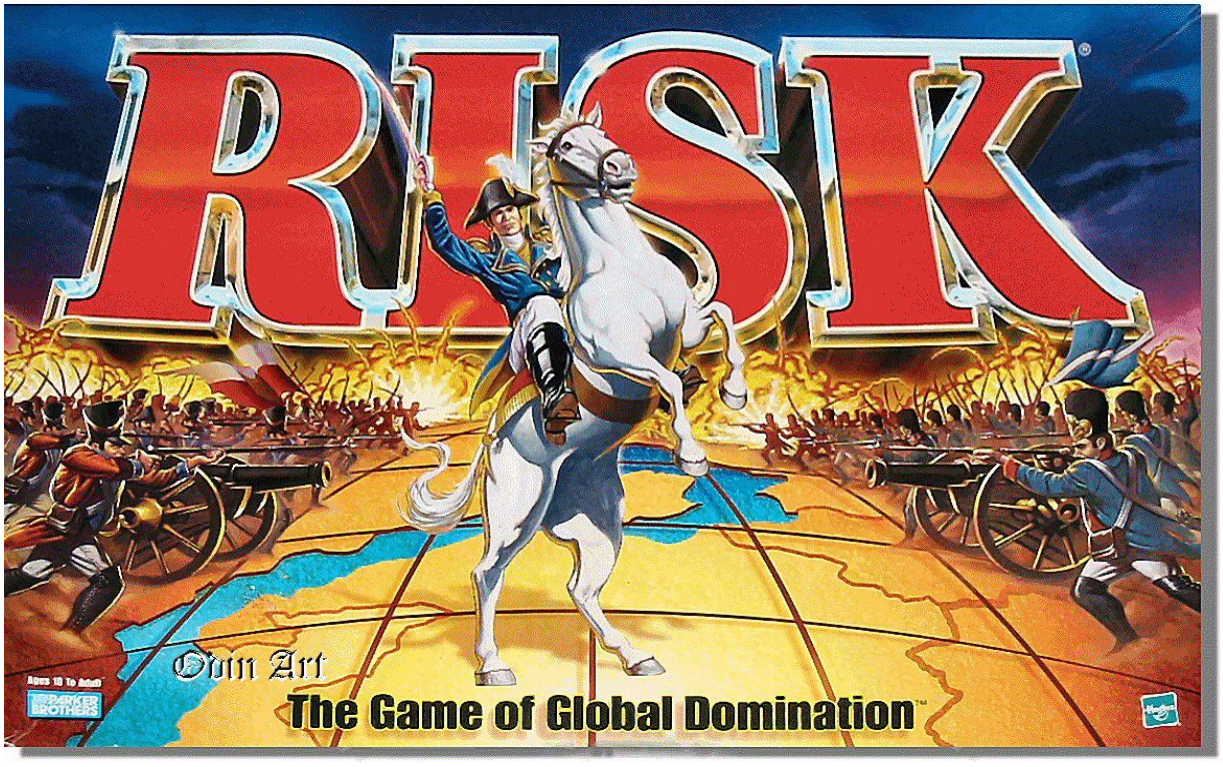


Figure 1: General Risk Game Cover[1]

2. Overview

2.1 General Information about Game

RISK is a famous military strategy game that fits for all ages. It has many different versions but we are using the standard version which is designed for three to six players and has a total of forty-two territories in six continents.

2.2 Armies

RISK has six different complete set of different colored army pieces and each of them has three units; the first one is Infantry, the second one is Cavalry (worth 5 Infantry) and the third one is Artillery (worth 10 Infantry, or 2 Cavalry).

2.3 Game Board

Figure 2: Game Board Map[2]

On the game board, there is a world map which is divided into six continents and forty-two territories. Each continent is represented in different colors and has 4 to 12 territories. Africa is represented in color brown and has 6 territories. Asia is represented in color green and has 12 territories.

Australia is represented in color grey and has 4 territories. Europe is represented in color blue and has 7 territories. North America is

represented in color yellow and has 9 territories and lastly, South America is represented in color orange and has 4 territories.
Some territories are adjacent to each other, either by a sea line or by sharing a border. At the bottom of the game board, there is a box which shows the number of Infantries that will be received by the player at the beginning of the turn.

2.4 Playing Pieces

Six different colors and 3 types of playing pieces. Infantry is one military unit, cavalry 5 military unit and artillery is 10 military unit for reducing the crowded. Moreover, there is one golden cavalry which belongs to no player and it is used to count bonus reinforcement.

Figure 3: Pieces of the Game[2]

2.5 Dices

Figure 4: Dices of the game[3]

In RISK, dices are used when attacking and defending Territories. There are 2 colors of dice, red and white and there are 3 red dices which are used to attack and 2 white dices which are used to defend.

To attack; the player can choose a territory to attack according to sea and boundary lines. Before rolling the dices, both of the players have to announce the number of dice they are going to use and have to roll at the same time. The player who is defending has to keep at least one soldier in her or his territory and after the attacker can roll 1, 2 or 3 red

dice. If the attacker rolls more dice, the chance of winning will be increased. The defender can roll 1 or 2 white dice. To be able to roll 2 dice, the relevant territory has to at least have two soldiers.

The biggest valued red dice is fit for the biggest valued white dice, same goes for the second biggest valued red and white dices. If the values are equal, the defender wins and the attacker lost a soldier, if the red dice's value is bigger the defender lost a soldier, until the defender or attacker lost all her soldiers, the attacker can keep attacking.

2.6 Cards

There are 2 types of cards; one of them represents territory cards and the other one represents mission cards. Mission cards are used only when playing RISK with the mission option. For territory cards, every single territory has one single card which has name and picture of the territory and one of the images of Infantry, Cavalry or Artillery. They represent how many soldiers can the player have at the beginning of the game.

Figure 5: Mission and Territory Cards of Risk Game[4]

2.7 Object Of the Game

The aim of RISK is to control all 42 territories while protecting your own territories and armies.

2.8 Claiming Territories

The player with the highest dice roll claims the first territory. To claim any territory, take one of your starting Units and place it into an empty territory of your choice. Now this territory is your own. The next player then places one of their Units into an empty Territory, claiming the Territories. To continue this until all 42 territories have been claimed. The player can not put a unit into a territory which is already claimed.

2.9 Reinforcing Territories

After all the territories are claimed you can start reinforcing them. If you have more units in one territory it is easier to defend that territory and launch attacks from.

There is no limit to the number of units that can occupy a territory, but each territory must contain at least one unit. You can choose to reinforce one territory with a large number of units or you can spread your units out across all of your territories.

2.10 Attacking and Defending

The aim of an attack is to capture a territory by defeating all the opposing units on it. For attacking,

- You may only attack a territory that's adjacent from border lines or sea lines.
- You must always have at least two armies in the territory you're attacking from. One army for defending your own territory and other one for attacking.
- You may continue attacking one territory until you have eliminated all armies on it, or you can continue attacking until you have one infantry. Also, you can shift your attack from one territory to another, attacking as many territories as you like during one turn.

2.11 End Turn

Players can choose to end their turn without doing any action or can do all other actions such as claiming, attacking or defending a territory or move their units based on their conditions.

2.12 Game Modes

In our game, we can play the game 3 to six player. And there is a two game mode: Everyman For Himself which is playing single and Tag Team which players playing as team of two person.

2.13 Leaderboard

This is a board which shows every single players' territories and army numbers.

2.14 End of the Game

The winner is the first player to eliminate every opponent by capturing all 42 territories on the board. All quests of one player completed.

3. Functional Requirements

In our design; a manual including the RISK game board rules will be provided in the main menu. In this way, we aim to help players to understand the game better.

Moreover, our game provides two gameplay mode; solo play and group play. So that players can choose to play as a group or enjoy the game in solo mode.

In the gameplay screen, our design provides a leaderboard which shows the unit numbers of every player. It is aimed to show the progress of the game to the players with this feature.Also, a scoreboard will be added to the main menu screen to show the highest scores.

Of course, the player will be able to set the sound and music on and of both from the main menu and the gameplay screen. A theme change in the background will be possible to create a more favorable background for the player herself.

In our design, we decided to not create units like cavalry and artillery, only infantries, because contrary to the board game, we have enough space for every Infantry in the game since we don't represent them on the world map as images but numbers. So we decided to eliminate the unit system and move on with only infantries.

3.1 Additional Requirements

In our design; we plan to add a new continent probably called "Oldies But Goldies Republic" on to the Atlantic Ocean between Europe and America which has sea-way connections to them. In this way, we plan to attribute to the strategic structure of the game by creating new connections between continents while changing the map.

Also, we plan to add a timer to count during turns. In this way, our game will become more fun to play because reducing the time of the turns will make the game faster and more fluent. We decided to make a normal turn 60 seconds long. On the other hand, we plan to add a fast turn, once in 10 turn the current player will have 30 second to finish her/his turn. Even if the player does not do anything, at end of the count, player's turn will be over.

We plan to add new quests in the mission cards in our design. Again, in this way, we aim to change the strategy dynamics of the game a little bit by creating new ways(missions) to win the game

Plus we thought about adding four seasons to game but ended up planning to add only the winter season to create more dynamics on the battlefield. In this new feature; for one full tour, a continent will have be

in the winter season and every battle on that continent, the attacker will have disadvantages on dice dynamics, so that the attacker will have less chance to win the battle.

Lastly we decided that adding the change of having epidemics in the soldier units in a random chance can change the dynamics of the winner or the loser part of the game. With this feature, when in a battle one of the sides can have epidemic in their army and if so the number of their soldiers on the battlefield will be decreased.

4. Non-Functional Requirements

4.1 Usability

In order to improve the usability of risk game, we focused to make the game more intuitive and satisfactory. When we examined other similar games (like Sid Meier's Civilization), intuitive gameplay is provided with buttons and commands that are denoted explicitly and with common names or visual elements (like icons, labels) are designed in a catchy and directive way. In this project, we followed the same aspect. Buttons are named shortly, explicitly and readable. Color schemes are context related. Instead of light, pastel colors that represent happiness and joy, dark matt colors are dominant.

Themes are added to make the game more customizable (such as Light, Dark, World War 2 theme). Thus, players can configure the game in their own way.

The risk game is supported by sound and visual effects. When the player clicks a button or starts a fight, the game will react visually or with a sound.

4.2 Reliability

Crashes are inevitable and affect the game experience. For reliability concern, we added save game system to the game. Even if there is a crash or freeze, the player can restart the game from last saved point. Thus, game progress will be secured. Also, there will be an auto-save system that saves game progress every 10 minutes. If the

player don’t save game and a crash occurs, he/she can continue from auto-save.

4.3 Performance

According to our tests, resources that risk game needs are approximately:

-256 MB RAM
-100-120 MB storage space
-Windows (XP and higher) OS
-Sound card
-Broadband internet connection

4.4 Installability

There is no need for a setup process for risk game. All information is stored in a jar file. The game can be executed from only this jar file. The game can be transferred with save files by only transfer this jar file

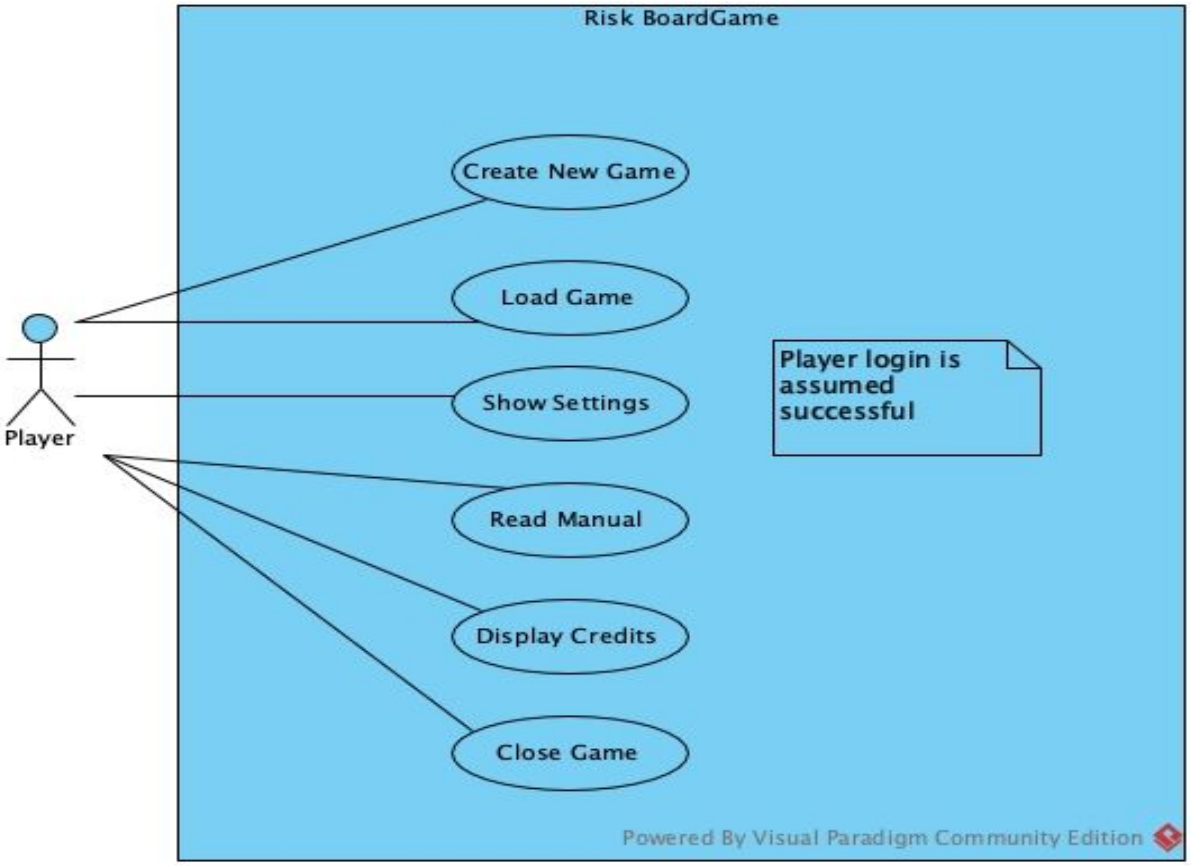
4.5 Additional Requirements

- The system that risk game run should have Java software.
- Codes are commented explicitly and objects are named informative.
- Time complexity is decreased to O(n).

5. System Models

5.1 Use Case Models

5.1.1 Menu-Use Case



Use Case 1 : Create New Game

Actor : User

Pre-condition : User has to be in the main menu

Entry Conditions : User has to select the number of players
User has to select the game type (Every man on himself or Grouping)

Exit Conditions : User has to select return menu on the screen which returns the user to main menu
In EMOH selection, one player wins the game
In Grouping selection, one group wins the game

Success Scenario Event Flow :

User selects number of player
User selects game type
Game starts according to given selections

Alternative Event Flow :

User selects 'Return Menu' on the screen and returns to main menu

Use Case 2 : Load Game

Actor : User

Pre-conditions : User has to be in the main menu
A saved game must be exists in the game's database

Entry Condition : None

Exit Conditions : User has to select return menu on the screen which returns the user to main menu
In EMOH selection, one player wins the game
In Grouping selection, one group wins the game

Success Scenario Event Flow :

Saved game starts according to last save

Alternative Event Flow :

User selects 'Return Menu' on the screen and returns to main menu

Use Case 3 : Show Settings

Actor : User

Stakeholders and Interests:

Player can play or mute the game sound
Player can play or mute the background music
Player can select one of the listed theme

Pre-condition : User has to be in the main menu

Post-conditions : Availability of the game sounds are updated
Availability of background music is updated
Theme is updated

Entry Conditions : User can select ‘Control Sound’
User can select ‘Control Music’
User can select ‘Choose Theme’

Exit Condition : User has to select return menu on the screen which returns the user to main menu

Success Scenario Event Flow :
User can control sound
User can control music
User can choose theme
User selects ‘Return Menu’ on the screen and returns to main menu

Alternative Event Flow :
User selects ‘Return Menu’ on the screen and returns to main menu

Use Case 4 : Read Manual

Actor : User

Pre-condition : User has to be in the main menu

Entry Condition : None

Exit Condition : User has to select return menu on the screen which returns the user to main menu

Success Scenario Event Flow :
A manual of the game is displayed on the screen
User selects ‘Return Menu’ on the screen and returns to main menu

Alternative Event Flow :
None

Use Case 5 : Display Credits

Actor : User

Pre-condition : User has to be in the main menu

Entry Condition : None

Exit Condition : User has to select return menu on the screen which returns the user to main menu

Success Scenario Event Flow :

Credits are displayed on the screen
User selects 'Return Menu' on the screen and returns to main menu

Alternative Event Flow :

None

Use Case 6 : Close Game

Actor : User

Pre-condition : User has to be in the main menu

Entry Condition : Exit Selection

Exit Condition : None

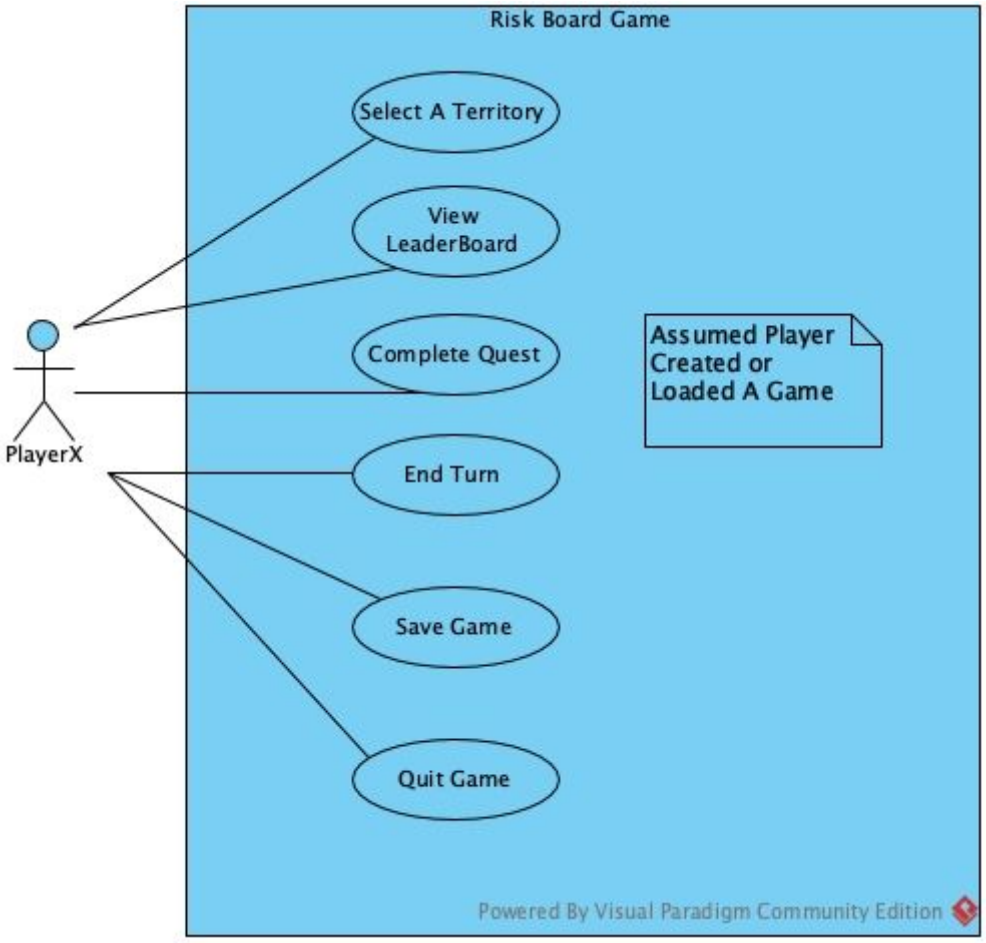
Success Scenario Event Flow :

User exits the game and the game screen is closed

Alternative Event Flow :

None

5.1.2 Create New Game-Use Case Model



<u>Use Case 1 : Select a Territory</u>

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User can choose to recruit her units
User can choose to transport her units
User can choose to attack her neighbours
User can choose to end her/his turn

Exit Conditions : User has to enter her ID and password or user has to create a new account to enter into menu page

Success Scenario Event Flow :
User enters her ID and password or creates a new account

Alternative Event Flow :
User has to select exit window to close the login page

Use Case 2 : View LeaderBoard

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User has to click the leaderboard icon on the window to see the scores

Exit Conditions : User has to click 'X' to return to the game window

Success Scenario Event Flow :
Scores are demonstrated on a panel

Alternative Event Flow :
None

Use Case 3 : Complete Quest

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User has to click the 'Complete Quest' to notify the game and other players that she/he completed her/his one quest

Exit Conditions : User has to click 'X' to return to the game window

Success Scenario Event Flow :
Current player's one mission is completed

Alternative Event Flow :
If the quest completion does not match with the current condition of the game, an error message will be displayed

Use Case 4 : End Turn

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User has to click the 'End Turn' to end her turn

Exit Conditions : User has to click 'X' to return to the game window

Success Scenario Event Flow :
Current player's turn is ended and now it's next player's turn

Alternative Event Flow :
None

Use Case 5 : Save Game

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User has to click the 'Save Game' to save the game progress

Exit Conditions : Game will automatically save the progress and user will stay on the gameplay window

Success Scenario Event Flow :
Current game progress is saved

Alternative Event Flow :
None

Use Case 6 : Quit Game

Actor : User

Pre-condition : User has to be in the game play

Entry Conditions : User has to click the 'Quit Game' and then 'Okay' button to to exit the game and return to the main menu window

Exit Conditions : None

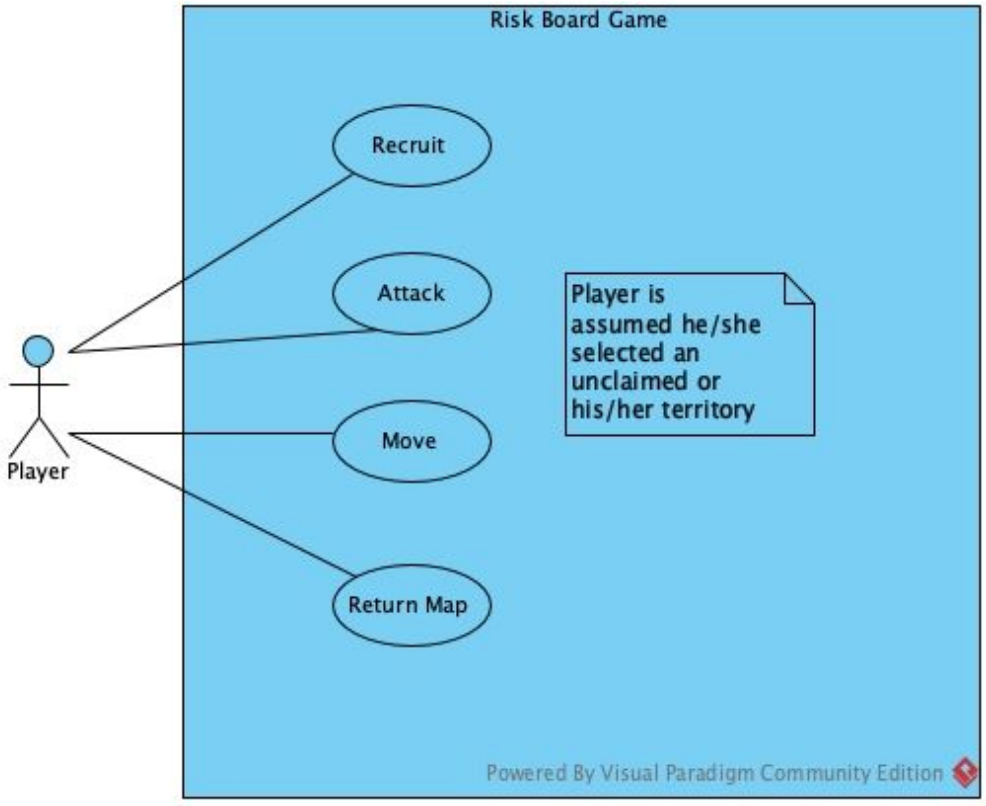
Success Scenario Event Flow :

User exits the game and return to the main menu window and if game has not been saved, game progress will be lost

Alternative Event Flow :

User clicks the 'Cancel' button to cancel the exit game choice and stays on the gameplay window

5.1.3 Play Game Use Case Model



<u>Use Case 1 : Recruit Unit</u>

Actor : User

Pre-condition : A territory should have been chosen by the user

Entry Conditions : User enters the troop amount to recruit

Exit Conditions : User returns to gameplay window automatically

Success Scenario Event Flow :

User enters the troop amount to recruit

Alternative Event Flow :

User chooses an impossible amount of soldier to recruit and an error page is displayed, then user is asked again to enter a troop amount

Use Case 4 : Attack a Neighbour

Actor : User

Pre-condition : A territory should have been chosen by the user

Entry Conditions : User chooses to attack a neighbour territory
User selects a neighbour territory to attack

Exit Conditions : None

Success Scenario Event Flow :

User attacks a neighbour territory

Alternative Event Flow :

User chooses a target land which is not a neighbour to the chosen territory and an error page is displayed, then user is asked again to select a target land

Use Case 3 : Move Units

Actor : User

Pre-condition : A territory should have been chosen by the user

Entry Conditions : User transports her units
User selects the target land to transport units

Exit Conditions : None

Success Scenario Event Flow :

User transports her units to the target land

Alternative Event Flow :

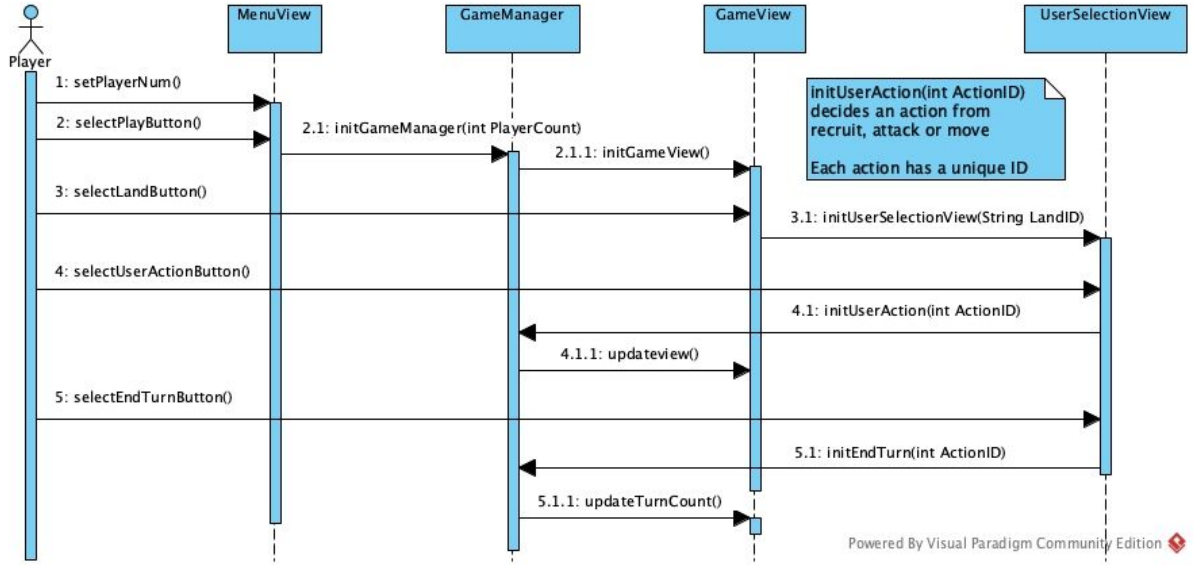
User chooses a target land which is not a neighbour to the chosen territory and an error

page is displayed, then user is asked again to select a target land

5.2 Dynamic Models

5.2.1 Sequence Diagrams

Beginning Of The Game



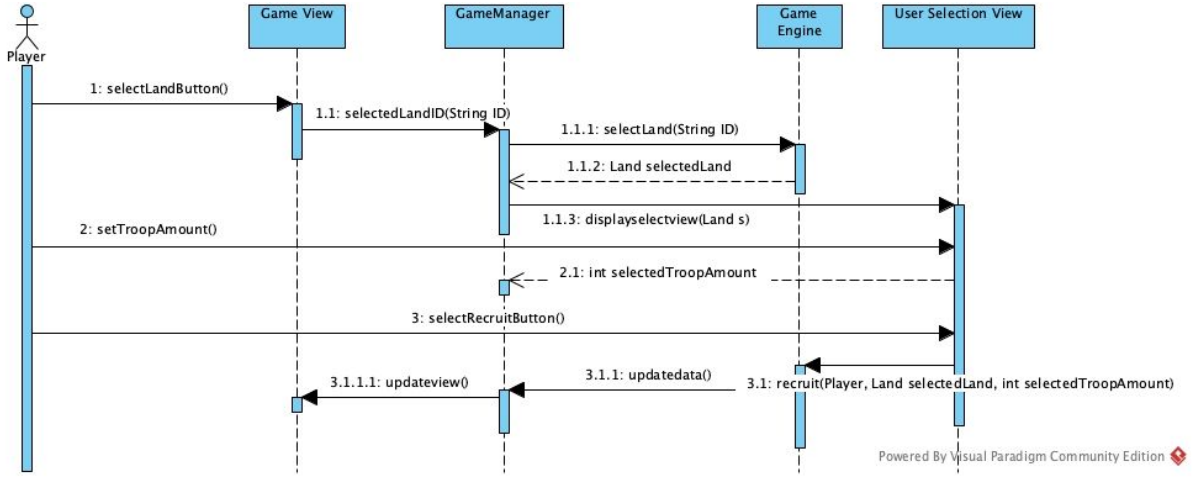
The purpose of the diagram is to show the interactions between the objects when a new game is created and played one turn.

Scenario

In this scenario Player creates a new game and plays his/her turn. Firstly player has to determine the number of players and select the Play Button in MenuView Frame. When Play Button is clicked a new GameManager object will be initialized according to player number. GameManager object will create a GameView Frame that displays the game board. There are buttons for each territory in GameView and users can select any button to take their actions. When a player selects a land button UserSelectionView frame will be created. This frame contains the actions that players can perform and each action has its unique ID. By selecting an action button the corresponding ActionID will be send to GameManager object. According to selected action the data in GameManager object will be modified and it will be updated on GameView. After performing actions user selects the End Turn button on User

Selection View. The turn count will be incremented on Game Manager and this player's turn will be over.

Recruit Action

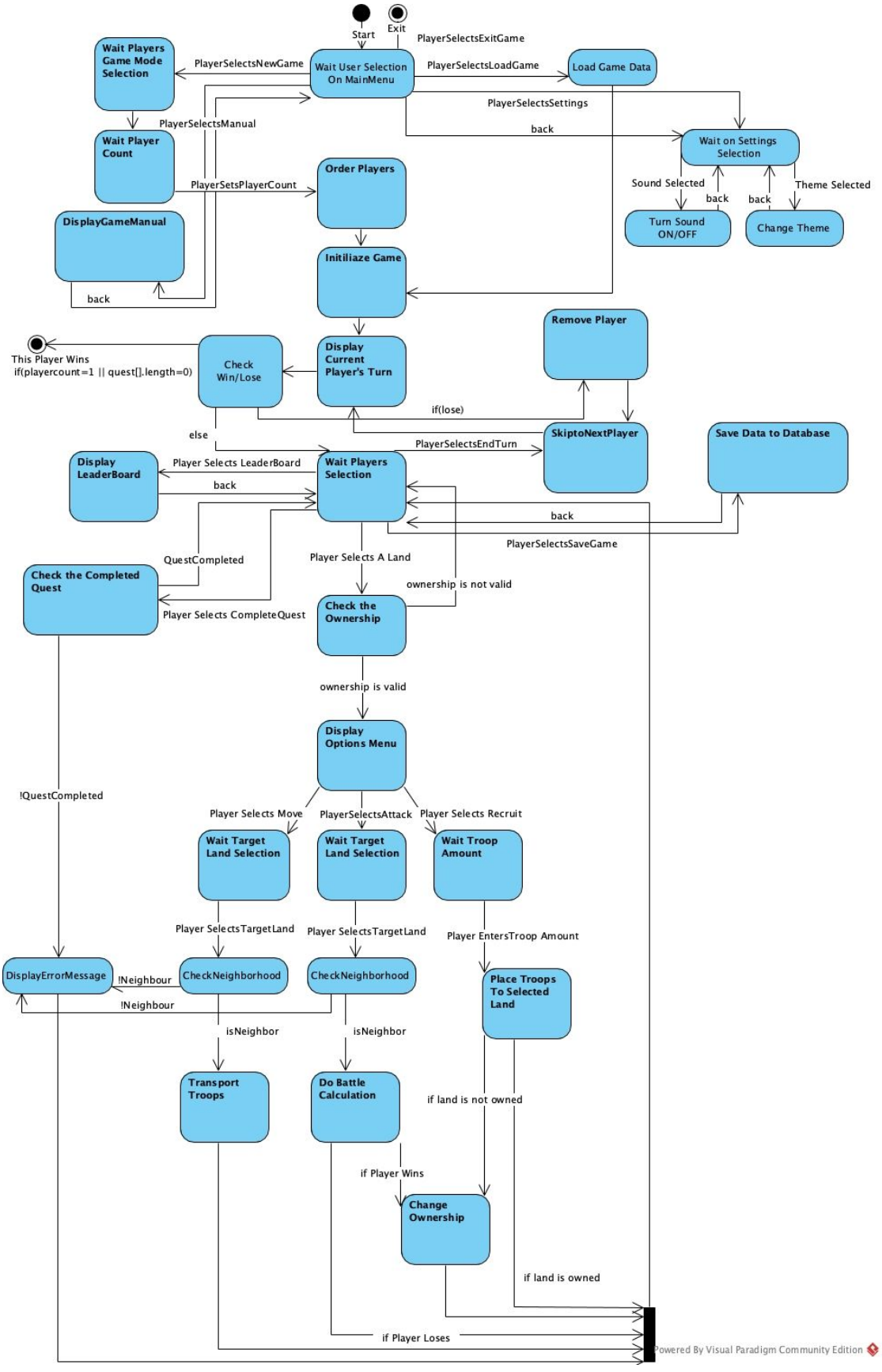


The purpose of this diagram is to show the interactions between objects when a user action is performed. Attack and Move actions follows a similar pattern.

Scenario

In this diagram Player selects a territory on GameView Frame. The selected landID will be send to GameManager object. In GameManager selectLand method of the Game Engine will be called to return selected Land object. Player has to choose the troop amount he/she wish to recruit before selecting recruit action. After selecting recruit button on UserSelectionView the selected troop amount will be sended to GameManager object and this object will call recruit method on GameEngine to perform recruit action. After recruit action performed the data in GameManager object will be modified and GameManager will update the GameView UI.

5.2.2 Activity Diagram



Explanation

In this diagram we assume that user login into game successfully. After login system will display the Main Menu View Frame and wait the user's selection. User can select Create New Game, Load Game, Show Manual, Settings.

Create New Game Selected

In this case system will ask the player number and game mode to user. By using these parameters system will initialize a new game. Before initialization system will also arrange the order of the players with respect to game rules. When the game initialized system will check the current player's win/lose conditions. In Risk if a player lose all of his/her troops he/she loses the game and removed from the player list. Unlike lose condition there are 2 win condition in Risk. First of them is becoming the last player in playerlist. The other condition is completing all assigned quests. These conditions will be checked by system before players actions and according to result system will decide that current player wins, loses or still in game. After that system will wait an input from the first player. According to selected user action such as recruit, attack or move system will check the actions correctness before performing any action. After performing selected action system will increment the turn count and check the next player's win/lose conditions.

If user selects save game, system will store game information to game database and wait next command.

Load Game Selected

If player selects Load Game in Main Menu system will initialize a game with the parameters in game database. After initializing the existing game system will follow the same pattern described on Create New Game Selected section.

Show Manual Selected

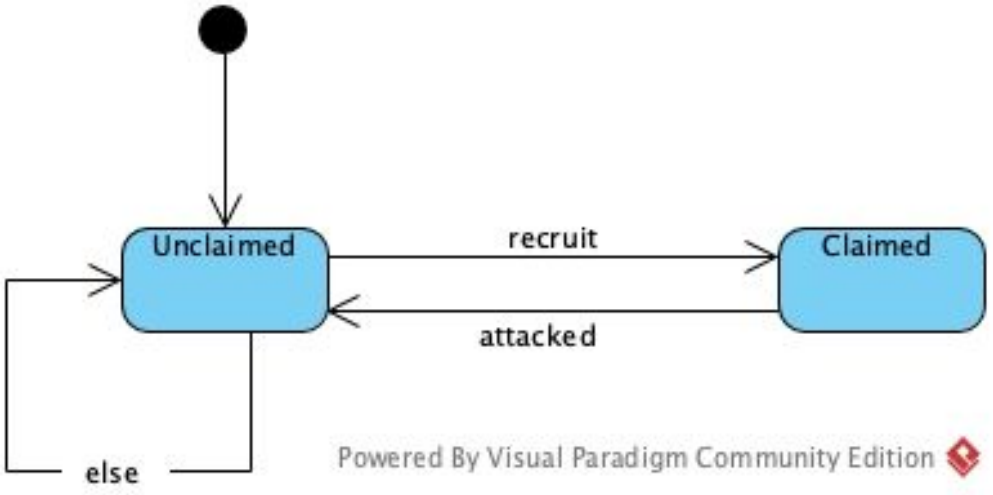
In main menu if show manual is selected system will display a new frame that contains game's rulebook and it will wait until return option is selected.

Setting Selected

In main menu if settings is selected system will display a new frame that contains theme and sound options. System will stay in this frame until player selects return option. System will also adjust the chosen UI settings

5.2.3 State Diagrams

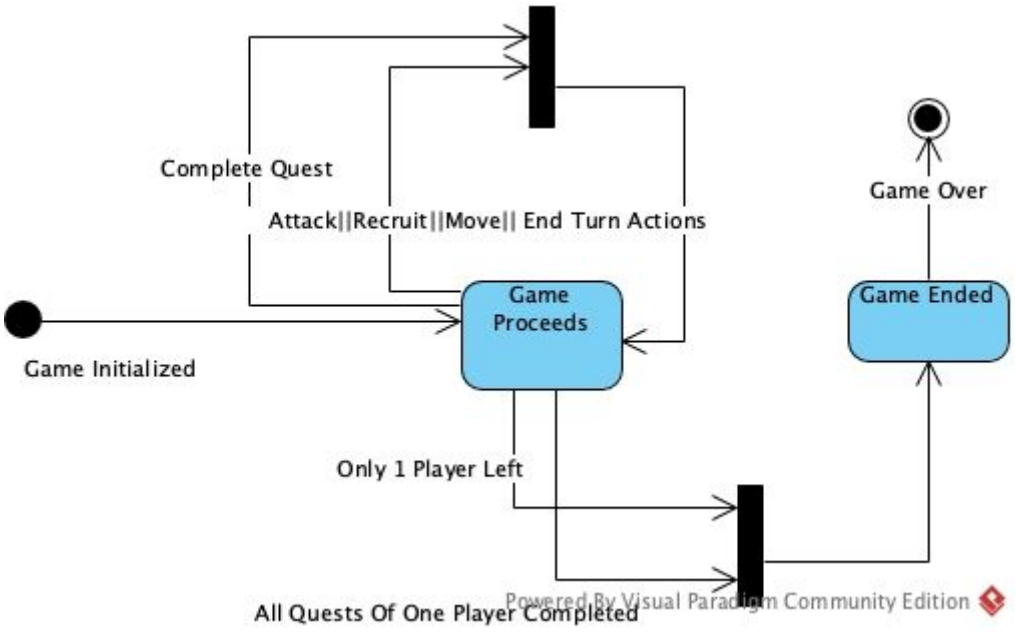
Land Object



Explanation

Land objects has 2 different state. In default all 42 lands start unclaimed. By using recruit action players may change this state to claimed. Players can change this state to unclaimed by performing a successful attack action on that Land.

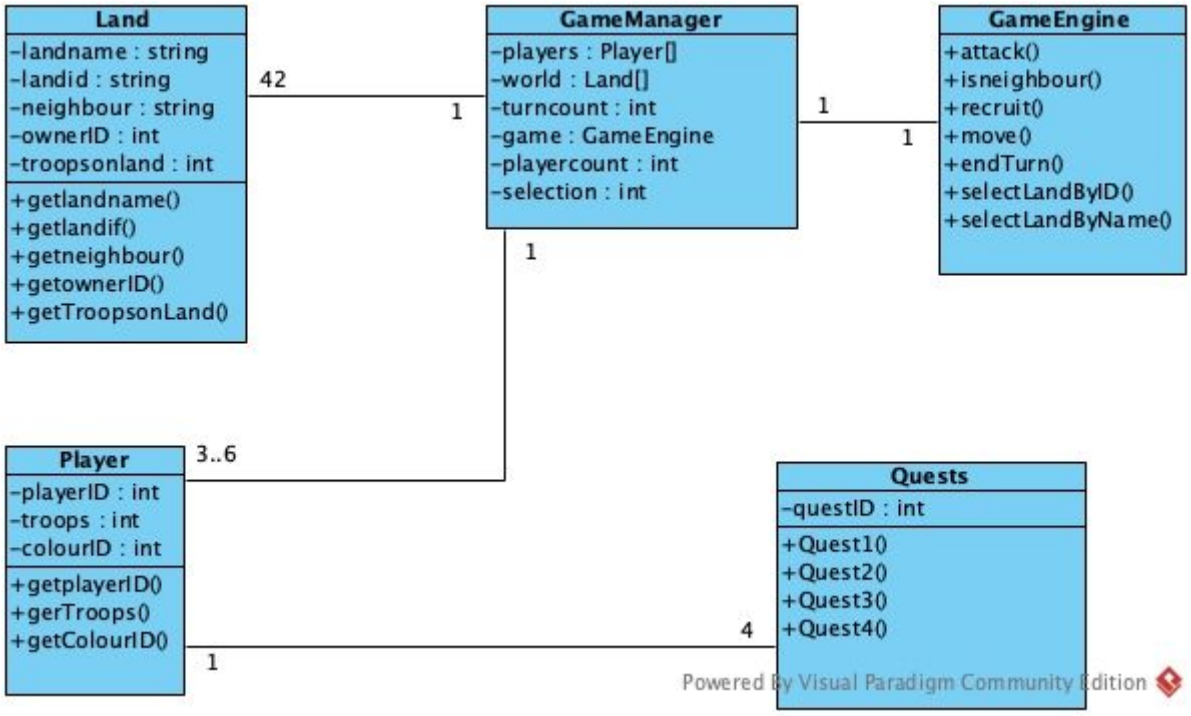
Game Object



Explanation

After initialization Game objects data will be modified constantly by player actions like recruit, attack, move, end turn and complete quest. In each iteration game object will check the number of remaining players and remaining quest amount on each player object. If there is only 1 player left in game or a player completed all of his/her quests game object will proceed to game ended state where the victory screen and credits is displayed. After Game Ended state the game data will be deleted and game object will be destroyed.

5.2.4 Class Diagrams



Explanation

This project consist of 5 main classes which are Land, Game Manager, Game Engine, Player and Quests.

Land

Land class has a constructor that takes landname, landID and neighbour parameters. At the beginning of the game all 42 land objects will be generated by Game Manager. Land class contains getter and setter methods in order to access and modify object's data.

Player

In a Risk game there are 3 to 6 players. These player objects are implemented by Player class. Player objects have their unique playerIDs, troop numbers and their unique color. Player class will be called by GameManager and by using getter functions their data will be modified.

Quests

Quest class consist of numerous methods that contains the logic of the quests such as tracking owned lands or successful battles. Each quest object will have a unique guestID. Each player object will contain 4 questID's and their progression on these quests will be checked by Quest class.

Game Manager

Game Manager is the main class in this project. It generates the player and land objects and it assigns 4 questID to each player object. After initialization Game Manager sends these objects to its game function where all game actions proceeds. In this function it calls methods from Game Engine according to players selections. Those methods updates the data in Game Manager and Game Manager updates the Game View Frame to display the changes.

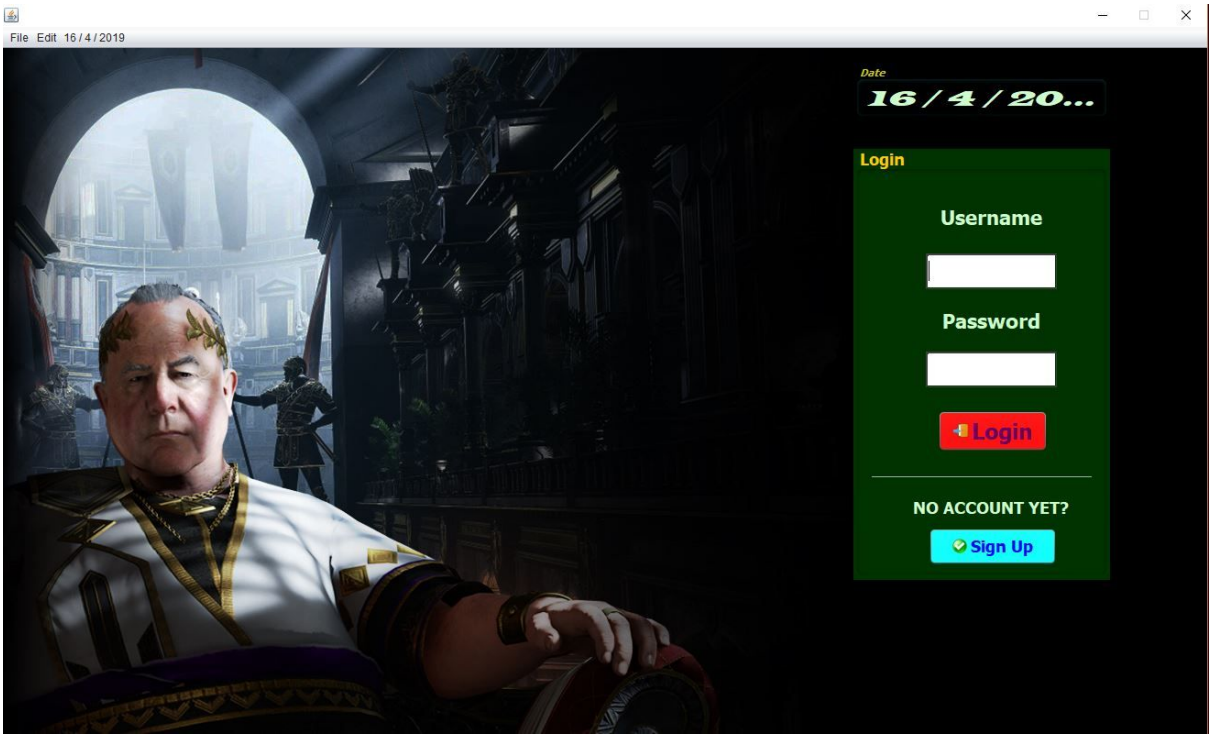
Game Engine

Game Engine class contains methods that perform game mechanics like recruit, attack and move. It also contains the endturn method that increments the turncount in Game Manager. Since Risk is a turn-based game turn count will be used to determine player's turns.

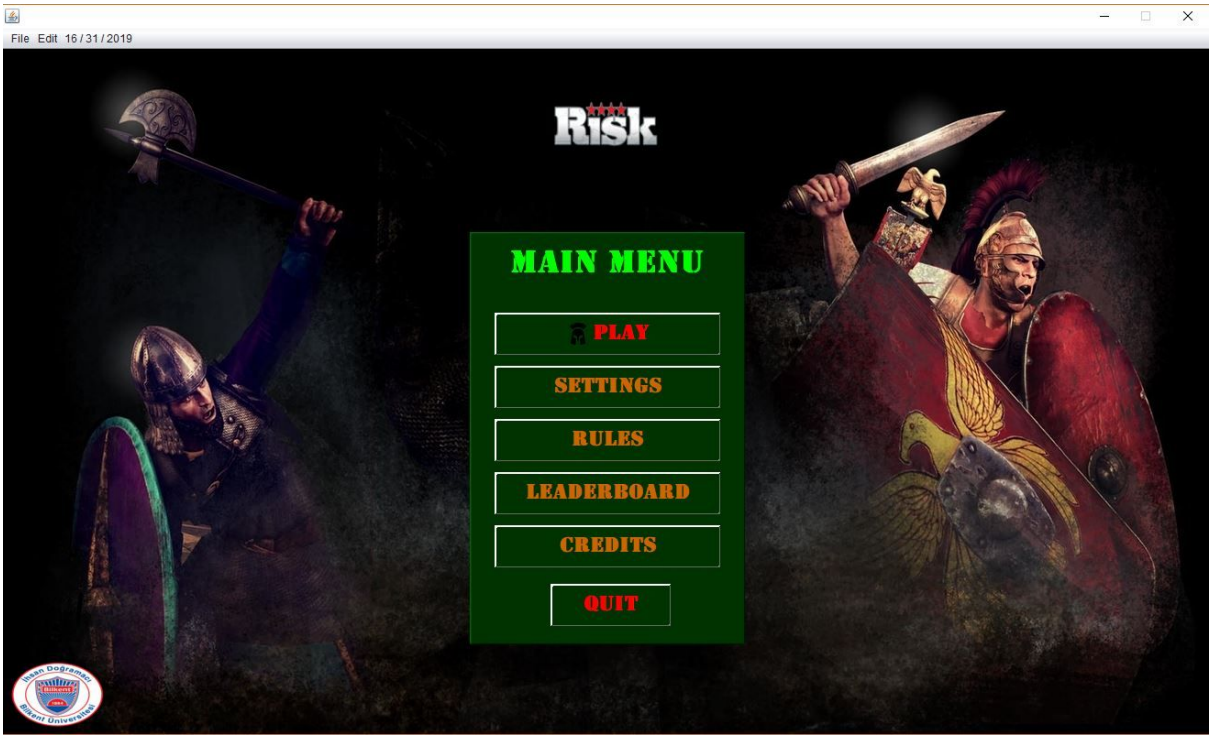
5.3 User Interface

5.3.1 Screen Mock-Ups

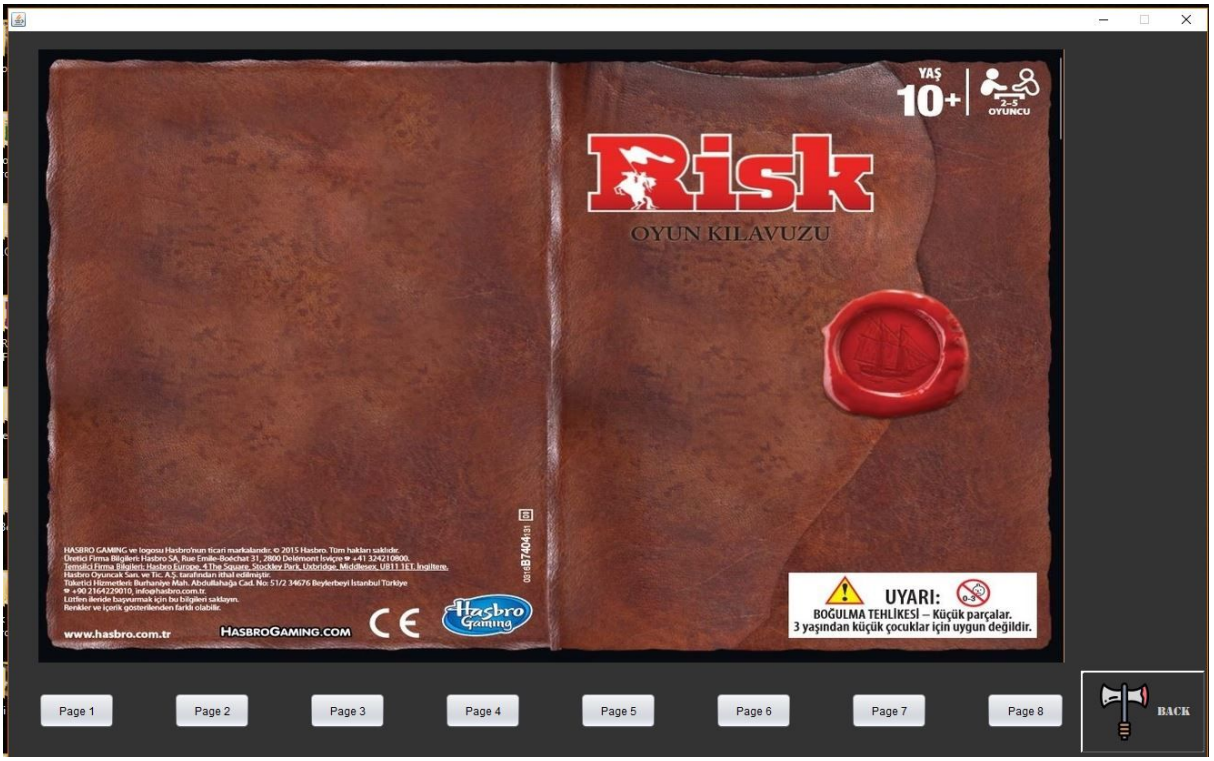
5.3.1.1 Login



5.3.1.2 Main Menu

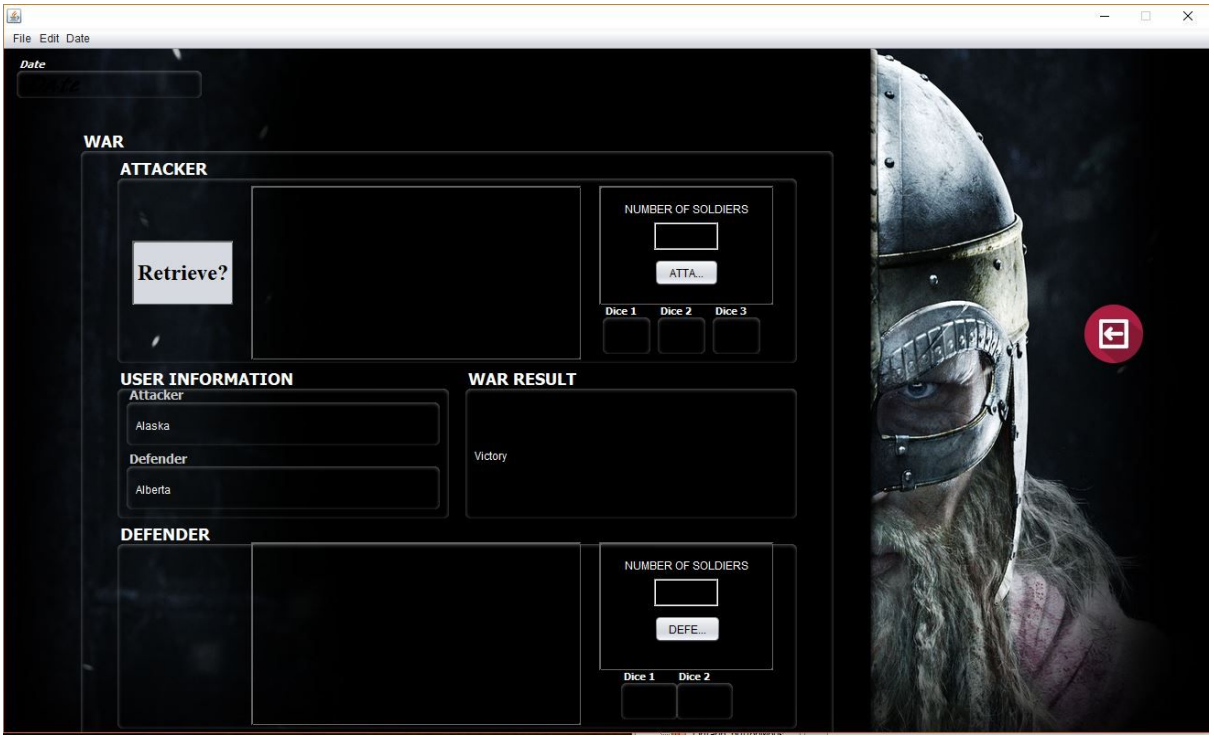


5.3.1.3 Tutorial

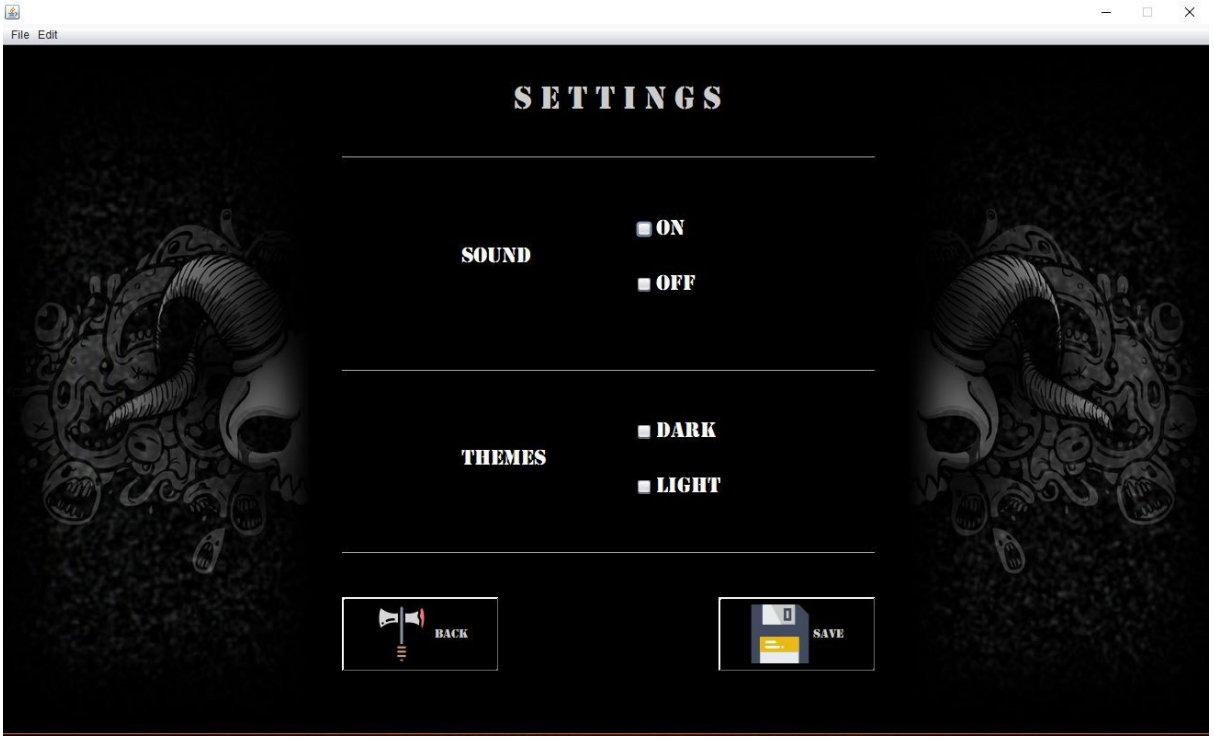


5.3.1.4 New Game

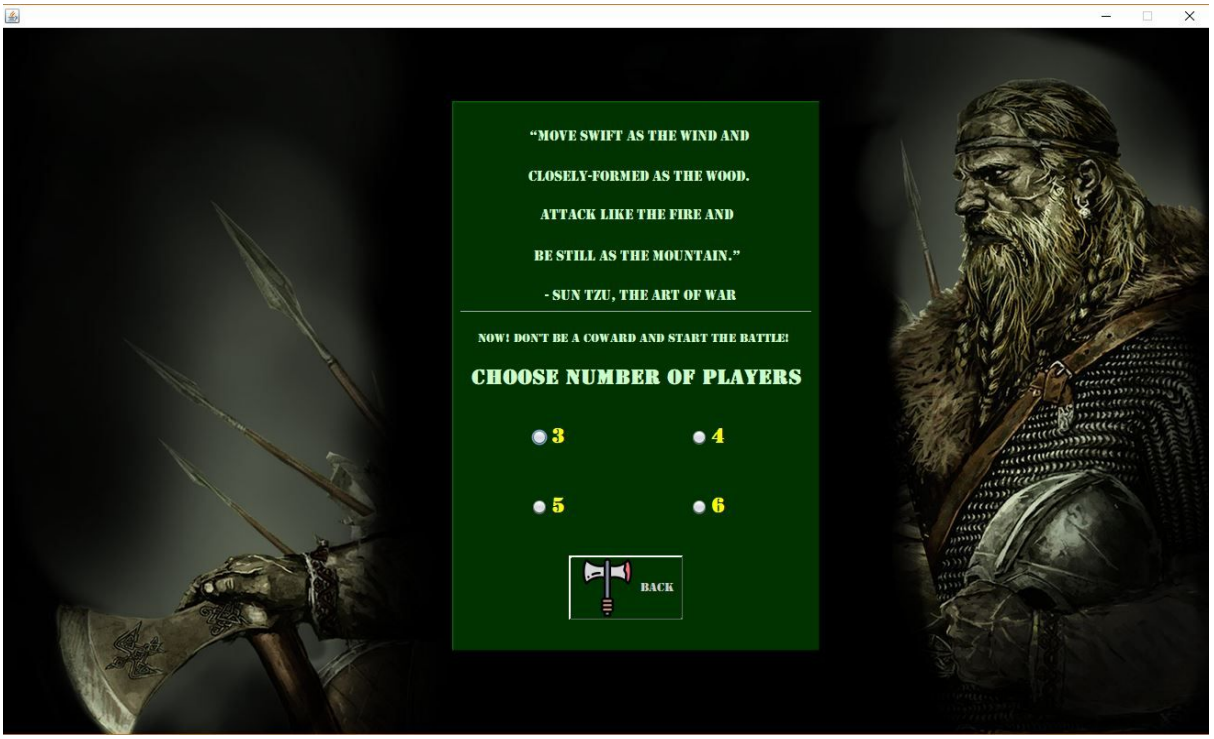
5.3.1.5 War



5.3.1.6 Settings



5.3.1.7 Number Of Players



6. Improvement Summary

As a summary, this virtual version of RISK will reflect all main specifications of the original game such as 3-6 players free-for-all mode, territory layout, cards etc In addition to these specifications, we will add some new features like background themes and sounds, leaderboard, tutorial of the game, tag team. To make the game more innovative we add a new continent which has 3 territories in the Atlantic ocean. We add a timer to make the game more interesting, the player can choose 2 different options. Moreover, we add new mission cards such as holding the middle east for 5 turns. Also, we add seasons will modify the dice results. Lastly, we add epidemics to reduce infantry count. All of these features make our game more interesting and unique.

The user interface is one of our main focuses. We want to design and implement this game suitable for all ages, because of this reason, we will keep UI as simple as possible but very effective at the same time to make players feel all the excitements of RISK.

7. References

[1]”Wikipedia”https://en.wikipedia.org/wiki/Risk_(game)[March, 2019]
[2]”UltraBoardgames”http://www.ultraboardgames.com/risk/game-rules.php[March,2019]
[3]”Boardgamegeek”https://boardgamegeek.com/boardgame/181/risk[March, 2019]
[4]”Boardgamegeekthread”https://boardgamegeek.com/thread/113195/drawing design-boardgame-pc-which-program-use[March, 2019]
[5]”Instructable”https://www.instructables.com/id/How-To-Design-Board-Games/[March, 2019]
[6]”Academia”https://www.academia.edu/7295897/Board_Game_Design_and_Implementation_for_Specific_Learning_Goals[April,2019]
[7]”Oracle”https://docs.oracle.com/javase/9/docs/api/javax/tools/package-use.html[April, 2019]