Bilkent University

Department of Computer Engineering



CS 319

Object Oriented Software Engineering *Game of “Risk”*

Group 1-E

Mehmet Ege Acıcan-21602186

Ali Ata Kılıçlı-21301530

Ekin Üstündağ-21602770

Mehmet Akif Kılıç-21502487

Seman Arda Yuvarlak-21501402

Contents

[1 Introduction 31](#_Toc254263637)

[2 Game Overview\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_31](#_Toc254263638)

2.1 Opening Screen\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3

2.2 Gameplay\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

2.1.1 Classical\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_4

2.1.2 Time Mod\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4

2.1.3 Conquer Specific Place\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4

2.1.4 Multiplayer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

2.3 Game Map\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 5

2.4 Game Troops\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6

2.5 Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6

2.5.1 Pre-Game-Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6

2.5.2 General Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

[3 Functional Requirements 71](#_Toc254263639)

**3.1 New Game, Saving & Slot Spaces\_\_\_\_\_\_\_ 7**

3.2 Arrangements of the Game\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

3.2.1 Pre-Game-Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 7

3.2.2 Gameplay\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

3.3 Settings\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

[4 Non-Functional Requirements\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 71](#_Toc254263650)

4.1 Usability\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

4.2 Reliability\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

4.3 Efficiency\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

4.4 Maintainability\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

4.5 Pseudo Requirements\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

[5 System Models 91](#_Toc254263651)

5.1 Use Case Diagram\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 9

5.2 Dynamic Models\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 19

5.2.1 Scenarios & Sequence Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_19

5.2.2 State Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 23

5.2.3 Activity Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23\_

5.2.4 Object and Class Models\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 25

5.2.5 User Interface\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_

6 References\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Analysis Report

Game of “Risk"

# Introduction

Risk is a strategy board game that is playable from 2 to 6 players. The main objective of the game is to eliminate other individuals from the game by conquering their occupied territories and have every territory in the world. In other words, World Domination!

Each player has their own troops, which can be in several different types, and territories. They distribute their troops to their territories and with these troops, they conquer other’s territories or defend their own territories. Also, players draw specific cards that allows them to gain specific amount of troops if they can catch a pattern with these cards such as three cards with the same troop type. The players gather and distribute these troops and they try to occupy as many different territories as possible in a turn. This process is repeated for every other player in every other turn. The game has an arbitrary length. It does not have a time limit.

In order to implement the game, we will use Java and JavaFX for framework.

# Game Overview

Game of “Risk” is the video game implementation of the board game itself, designed for laptops and desktops. While the basic core of the game does not change, the game tries to fix specific problems that the original game and adds specific updates such as increased amount of player. The game greets the player with a welcome message and then highlights its 4 gameplay options. After the gameplay option is chosen, the pre-game-settings menu appears. The settings menu allows how many players will play and which color will each player be. After the pre-game-settings menu, the game will start.

* 1. **Opening Screen**

The games opening screen will have 3 options: Play, Settings and Exit. Play will allow you to start a new game or load a previous game. Settings will allow you to change specific settings.

* 1. **Gameplay**

Game of “Risk” has got 4 game mods(4 different ways you can play the game): 1-Classical,2-Time Mod,3- Conquer specific places.

* + 1. **Classical**

This is the classical version of the Risk Board Game. The game is going to be played in the way the classical board game is played. In their own turns, each player will first receive specific amount of troops that they will distribute, then they will attack to the other territories, then they will redistribute their remaining troops after their conquest process. The conquests will be determined by dice rolls: Based on the number of the troops the territory, the attacker and the defender will throw dices and each higher result that the attacker has will result in decreasing of one troop. However, the number of players is extended. 7 players are able to play risk.

* + 1. **Time Mod**

This gameplay system is designed specifically to handle the major problem of the classical board game: Taking too much time. The classical game can take hours and in some cases, days. The pre-game-settings, the players adjust the time. The gameplay is same with the classical game mod however, the person with the most territories at the end of the time limit wins the game.

* + 1. **Golden Territories**

This pre-game-settings makes the players to decide specific places to take. Each different player can have different places for winning or each one of them can try to take the same place. The person who conquers their selected territories wins. Otherwise, the gameplay is the same with the classical game mod.

* 1. **Game Map**

The game map depicts the political map of Earth, divided into 42 different places. These 42 different places are the territories and players conquer these territories.



Figure 1: Game Map of Game of “Risk” [3]

# If the user adjust map to Tamriel in settings, the game map will be changed to the Tamriel world map, which is the world map of the famous fantasy game, *The Elder Scrolls.*

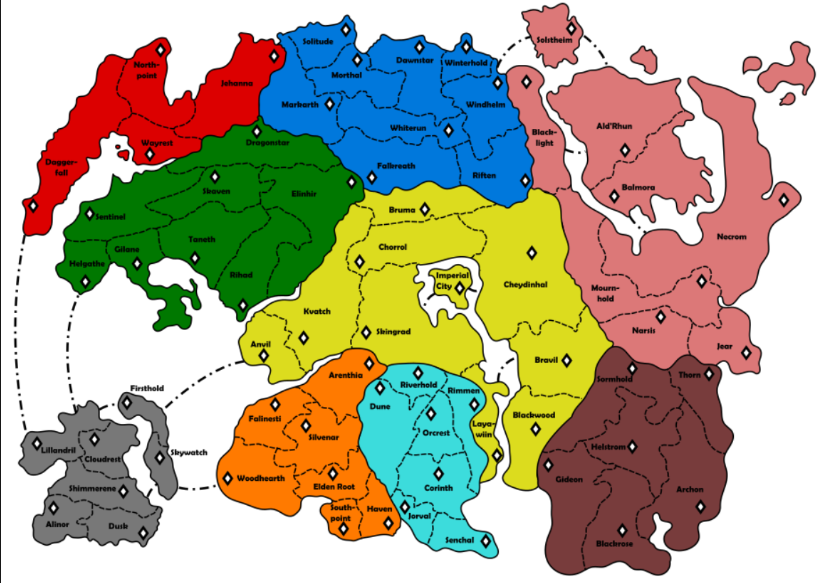


Figure 2: Additional Tamriel Map for Game of “Risk” [2]

* 1. **Game Troops**

Troops are the units that player uses to conquer territories. There are 5 types of units in Game of “Risk”, each with their own power ranking. Power ranking specifies how many troops does it take to destroy another troop. For example, it takes 5 soldiers to destroy 1 cannon.

|  |  |
| --- | --- |
| Troop Type | Power Ranking |
| Soldier | 1 |
| Cavalier | 3 |
| Cannon | 5 |
| Tank | 7 |
| Plane | 9 |

* 1. **Settings**
     1. **Pre-Game-Settings**

If the players are playing the game on Time Mod, the players are going to be able to adjust time.

If the players are playing the game on Golden Territory Mod, players are going to be able to adjust their own golden territories.

* + 1. **General Settings**

# The players are able to adjust the sound and the brightness of the game.

# Functional Requirements

* 1. **New Game, Saving & Slot Spaces**

New Game: Users should be able to start a new game, select a mod, arrange their troops, arrange the time( if the game is played on Time Mod) and arrange the golden territory( if the game is played on Golden Territory mod)

Game Saving: Users should be able to save the games they are playing and be able to continue any time that they want.

Slot Spaces: Users should have different slots for games so that they can save multiple games and be able to continue to the game that they choose from the slot.

* 1. **Arrangements of the Game**
     1. **Pre-Game-Settings**

Troop color: Before starting to the game, it does not matter which mod the players are playing, the users should be able to choose their troops color since each color offers different troop ability to the users.

|  |  |
| --- | --- |
| Troop Color | Effect |
| Red | Starts the game with extra 10 soldier troops |
| Blue | Starts the Game with extra 5 Cavalier troops |
| Yellow | Starts the Game with extra 2 Tank troops |
| White | Starts the Game with extra 1 plane troop |
| Orange | Starts the game with one extra territory |
| Black | Starts the game regularly |
| Green | Starts the game with extra 3 cannon troops |

Figure 3: Table for player colors

If the users are playing it on the Time Mod, the users should be able to adjust the time

If the users are playing on Golden Territory Mod, the users should be able to select the golden territories, if occupied the player wins type of territories.

* + 1. **Gameplay**

Troop Movement: The user should be able to drag a specific troop that they want to the desired territory for recruitment and distribution stages in their turns. The user should only be able to drag troops to the territories that they own. If the user tries otherwise, an error message should pop up, and the game should prevent it. After dragging, the soldier count(the sum of the power ranks of all troops) should be effected and displayed.

Territory owning: The game should display which user owns which territory clearly to the players. This can be done by multiple ways: Changing the colors of the territories or highlighting the troop colors in the specific region( if the red player has Alaska region, there should be a red point in the middle of Alaska Territory with the total power rank of troops in the region.

Timing: If the users are playing the game on Time Mod, A clock should show how much time is left for the game to finish.

Hints: If the users are confused about gameplay or specific aspects of the game, there should be a hint button to display helps to the player

Save: Whenever decided, the user should be able to save the game whenever he or she desires.

Exit: The User should be able to quit the game whenever decided.

* 1. **SETTINGS**

The users should be able to adjust the brightness of the game whenever they desire

The users should be able to adjust the volume of the game whenever they desire.

# Non-Functional Requirements

* 1. **Usability**

The interface is going to be user friendly as much as possible. Making the gameplay of the game easy, fluent and as fun as possible for the users. The user will be able to drag and drop troops with mouse.

* 1. **Reliability**

The game will have failure probability close to 1%. Our intension is that the game will work accurately without any crashes or bugs. Even if a crush occurs, the game will continue from where it stopped after reopening it.

* 1. **Efficiency**

The game will satisfy an efficiency constraint. This efficiency constraint is that, the game will handle user interactions as quickly as possible. This quickness term can be considered as:

User interactions:

1. Dragging the troops must take 0.01 seconds
2. Clicking a button for a specific action and seeing the outcome of the of it must take 0.01 seconds. For example, if the user wants to view the Hints, after the user clicks to the hints button, the hints should be open at 0.01 seconds.
   1. **Maintainability**

The implementation of the game will be in a style to allow easy understanding of it and allowing possible contribution to it. The Analysis and Design reports will be able to clarify the understanding of the project and its implementation even further.

* 1. **Pseudo Requirements**

For the implementation of the code, Java programming language will be used

For the User Interface, JavaFX will be used. We preferred JavaFX instead of Java because JavaFX offers a simpler and better User Interface than Java’s.

# System Models

* 1. **Use Case Diagram**

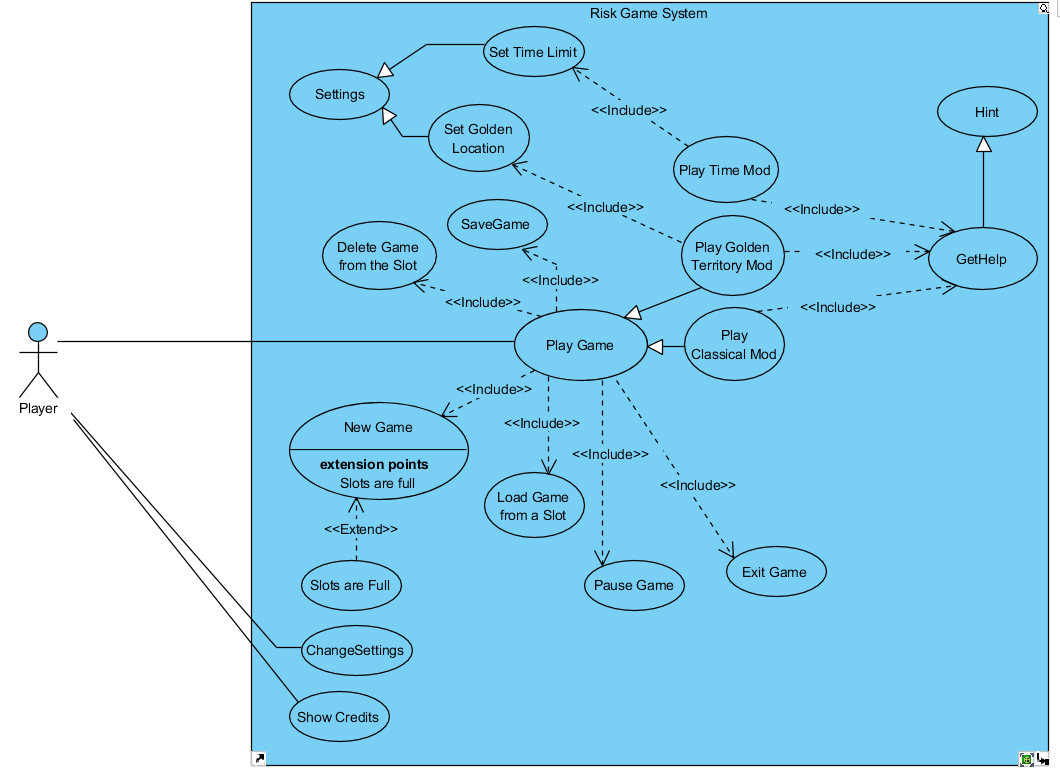


Figure 4: Use Case Diagram

Use Cases in the Use Case Diagram can be understood directly by their names and their relationships with other use cases. However, they require further explanation in order to be understood completely.

1)

|  |  |
| --- | --- |
| Use Case Name | GetHelp |
| Participating Actors | Player |
| Flow of Events | … |
| Entry Condition | Player plays any 4 of the game mods |
| Exit Condition | Player Got help |

2)

|  |  |
| --- | --- |
| Use Case Name | Hint |
| Participating Actors | Inherited from Get Help |
| Flow of Events | 1.The player clicks GetHelp button  2.Hint screen appears |
| Entry Condition | Inherited from Get Help Use Case |
| Exit Condition | Inherited from Get Help Use Case |

3)

|  |  |
| --- | --- |
| Use Case Name | Play Time Mod |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses Time Mod option  2.The time arrangement screen appears  3.Player adjusts the time  4.Players choose their colors  5. The game starts, arranged clock starts decreasing  6. The Game ends when clock hits 0 |
| Entry Condition | The player has chosen Play Time Mod |
| Exit Condition | The player exits the game prematurely  The result appears as “Player [number] has won!” |

4)

|  |  |
| --- | --- |
| Use Case Name | Set Time Limit |
| Participating Actors | Inherited from settings |
| Flow of Events | 1.The player chooses settings mod before starting the Time Mod Game  2. Time adjustment screen appears |
| Entry Condition | Inherited from Settings |
| Exit Condition | Inherited from Settings |

5)

|  |  |
| --- | --- |
| Use Case Name | Play Golden Territory Mod |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses Golden Territory mod option  2.The territory arrangement screen appears  3.Players decide which territories are the golden territories(Win Condition)  4.Players choose their colors  5. The game starts  6. The Game finishes when a player conquers his/her golden territories |
| Entry Condition | The player has chosen Play Golden Territory Mod |
| Exit Condition | The player exits the game  The result appears as “Player [number] has won!” |

6)

|  |  |
| --- | --- |
| Use Case Name | Set Golden Location |
| Participating Actors | Inherited from settings |
| Flow of Events | 1.The player chooses settings mod before starting the Time Mod Game  2. Golden Location adjustment screen appears |
| Entry Condition | Inherited from Settings |
| Exit Condition | Inherited from Settings |

7)

|  |  |
| --- | --- |
| Use Case Name | Settings |
| Participating Actors | Players |
| Flow of Events | … |
| Entry Condition | Player plays either Time Mod or Golden Territory mod |
| Exit Condition | Player either adjusted the time or set the Golden Territories |

8)

|  |  |
| --- | --- |
| Use Case Name | Play Classical Mod |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses Classical mod option  2.Players choose their color  3. The game starts  4. The Game finishes when a player conquers the world |
| Entry Condition | The player has chosen Classical Mod |
| Exit Condition | The player exits the game  The result appears as “Player [number] has won!” |

10)

|  |  |
| --- | --- |
| Use Case Name | Show Credits |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses Classical mod option  2.The game shows credits |
| Entry Condition | The player is on the main menu |
| Exit Condition | The player opened credits |

11)

|  |  |
| --- | --- |
| Use Case Name | Pause Game |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses to pause the game  2.The game stops |
| Entry Condition | The player is in the game |
| Exit Condition | Game displays pause menu(Resume-Save-Exit) |

12)

|  |  |
| --- | --- |
| Use Case Name | Exit Game |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses to exit the game  2.The game stops and it is not saved |
| Entry Condition | The player has paused the game |
| Exit Condition | Game stops without saving |

13)

|  |  |
| --- | --- |
| Use Case Name | Load Game from a Slot |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses to load game from a slot  2.The saved game opens and the game continues |
| Entry Condition | The player has clicked play game button |
| Exit Condition | The saved game opens and the game continues |

14)

|  |  |
| --- | --- |
| Use Case Name | Delete Game from a Slot |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses to delete a game from the slot  2.The saved game is deleted |
| Entry Condition | The player has clicked the play game button |
| Exit Condition | The saved game is deleted |

15)

|  |  |
| --- | --- |
| Use Case Name | Slots are full |
| Participating Actors | Player |
| Flow of Events | Game gives warning that there is not an additional saving slot, thus nonew game can be played |
| Entry Condition | Extends New Game use case. Player can not play a new game |
| Exit Condition | Player have received an error message on saving the game. |

16)

|  |  |
| --- | --- |
| Use Case Name | New Game |
| Participating Actors | Player |
| Flow of Events | 1.The player chooses to play the game  2.The 4 gameplay mods( Tmriel,Classic,Golden Territor and Time Mod) have been asked  3. The player chooses one of the 4 mods  4. Game is saved to a slot  5. Game starts |
| Entry Condition | The player has clicked play game button |
| Exit Condition | One of the 4 gameplay mods have been chosen and game has been saved to a slot |

17)

|  |  |
| --- | --- |
| Use Case Name | Delete Game |
| Participating Actors | Player |
| Flow of Events | … |
| Entry Condition | The player is playing the game |
| Exit Condition | The player has deleted the game |

18)

|  |  |
| --- | --- |
| Use Case Name | Save Game |
| Participating Actors | Player |
| Flow of Events | … |
| Entry Condition | The player is playing the game |
| Exit Condition | The player has saved the game |

* 1. **Dynamic Models**
     1. **Scenarios & Sequence Diagrams**
        1. **New Game or Load Game**

After the player clicks play game, Slot choosing Screen appears. In the slot choosing screen, the player can start a new game (If there are available slots) or the player can load a previous game slot. Then, the game loads the game game data by reading the save file. The game data consists of game map, troop data of the players and territory data of the players, if a new game has started, these datas are all in initial condition

* + - 1. **Select Gameplay Mods**

# If the user decides to start a new game, then 4 gameplay mods will be presented to the player. The player will choose one of the 4 mods, and then the game will save the decision to a save slot.

* + - 1. **Game Settings**

If the player has chosen Time Mod, a setting will appear to arrange the time limit. The player will adjust the time and the game will start with the clock arranged by the player

If the player has chosen Golden Territory mod, the setting will appear to arrange which territories are going to be golden territories. The player will adjust the golden territories and the game will start.

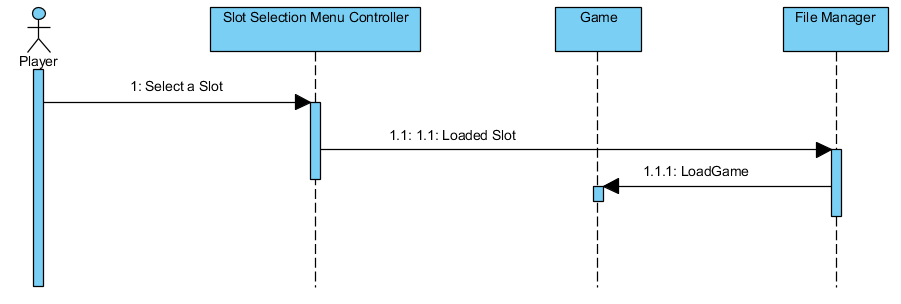


Figure 5: Sequence Diagram of Load Game

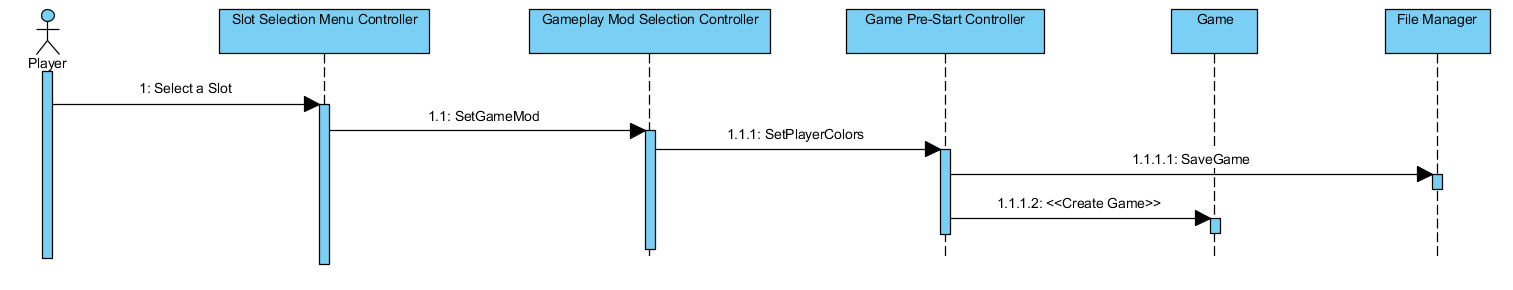


Figure 6: Sequence Diagram of New Game

* + - 1. **Classical Mod**

After creating a new game and choosing to play on Classical Mod, the player will come across with screen in which he/she has to choose the troop color which gives the player a different start. After this, the Game loads and starts. Firstly, The game gives troops to the players on their respected turn. Then, the player distributes their troops. The game checks whether the distribution is applicable. After this the player decides to attack specific locations of his/her opponents. The game checks whether the attack that they have decided is possible. After the attack stage, the distribution stage starts, then the game checks whether the distributions decided by the player is possible. The sequence diagram is given below:

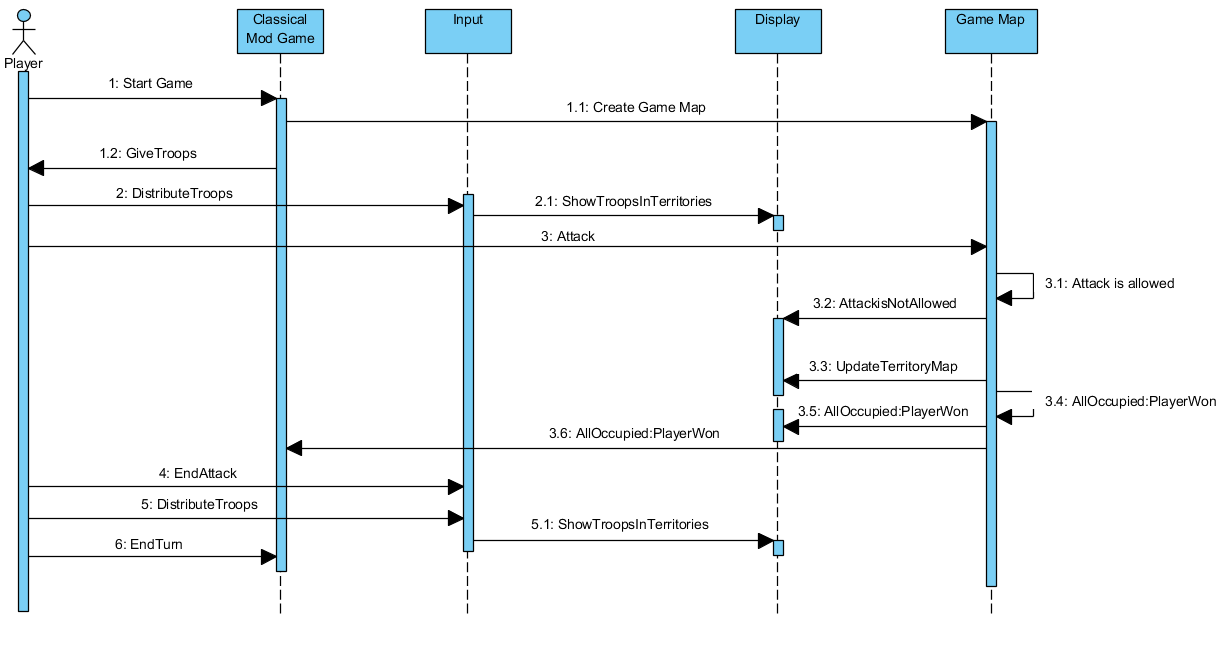


Figure 7: Classical Mod Sequence Diagram

* + - 1. **Time Mod**

The diagram is almost similar to classical mod diagram however, only with the addition of time countdown. The sequence Diagram is given below:

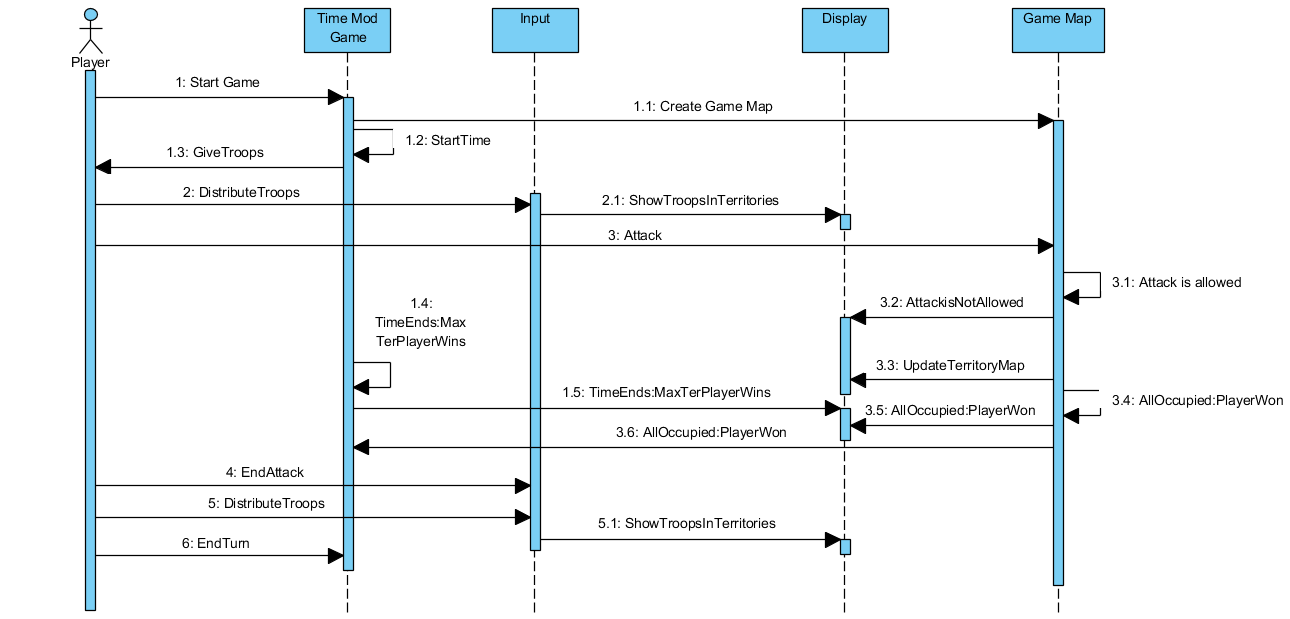


Figure 8: Sequence Diagram of Time Mod

* + - 1. **Golden Territory Mod**

Golden Territory Mod is identical to Classical Game Mod. The only difference is that instead of all territories, the map looks at golden territories

* + 1. **State Diagrams**
       1. **Classic Mod Gameplay State Diagram**

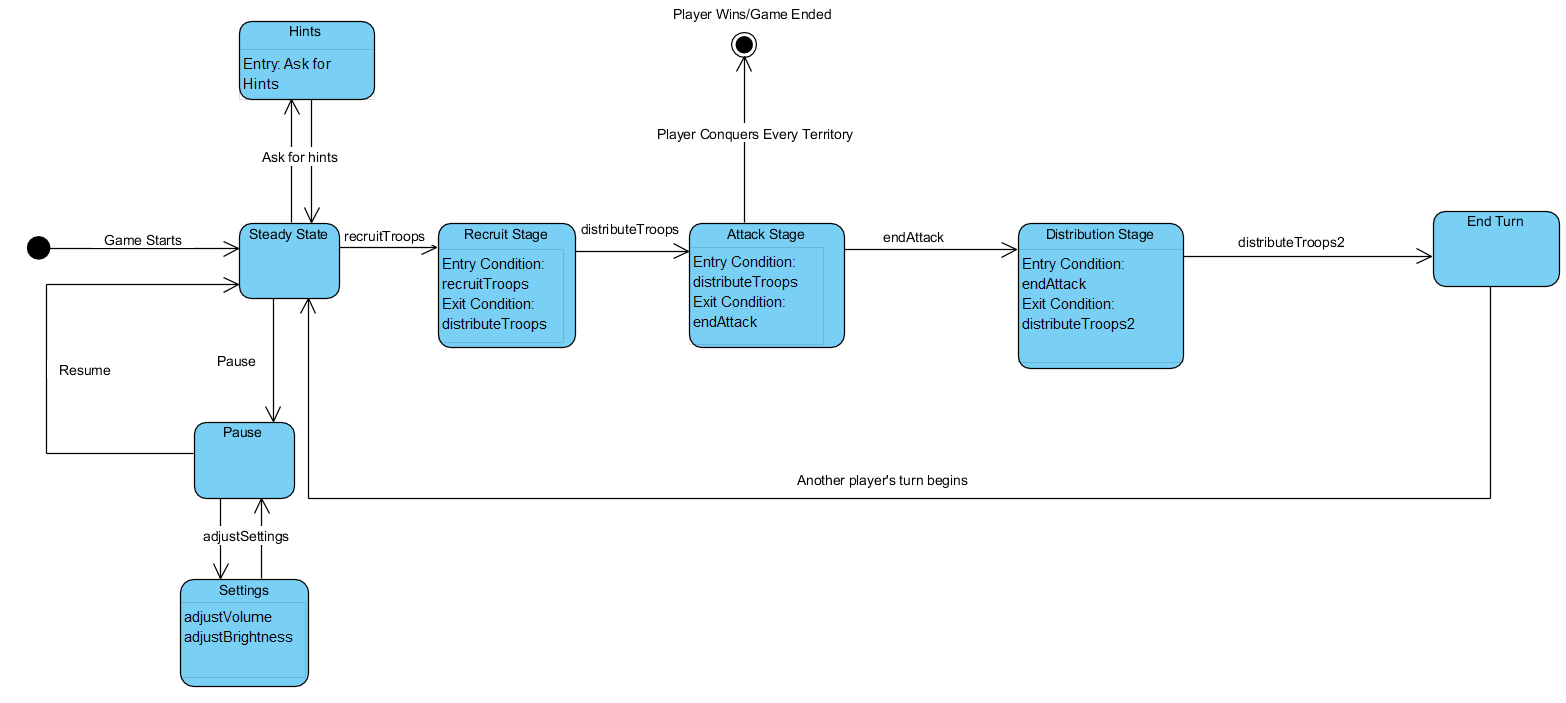


Figure 9: State Diagram of the Classical Mod

In classical mod, player starts in a steady state where the player is able to ask for hint and adjust the settings. Later, the stages of game begins. Firstly, player recruits troops, allowing him/her to enter to Recruit Stage, then the player distributes his/her newly gained troops. Later, the Attack Stage begins, in this stage the player tries to occupy as many territories as possible, if the player occupies all territories, he/she wins the game. Then the player ends the Attack stage. This allows the beginning of the distribution stage which the player distributes his/her remaining troops. Then the player ends the turn and another players turn begins. The same process continues for another player.

* + - 1. **Time Mod State Diagram**

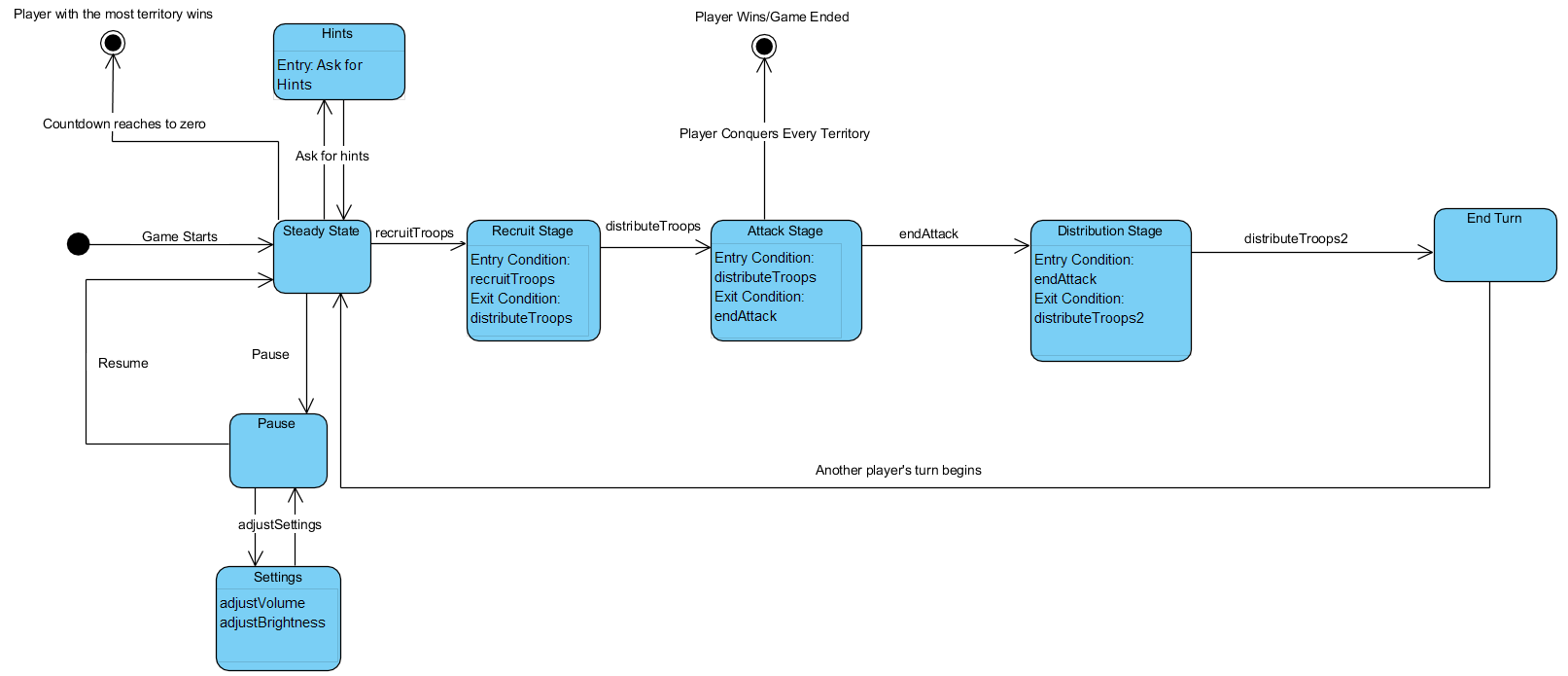


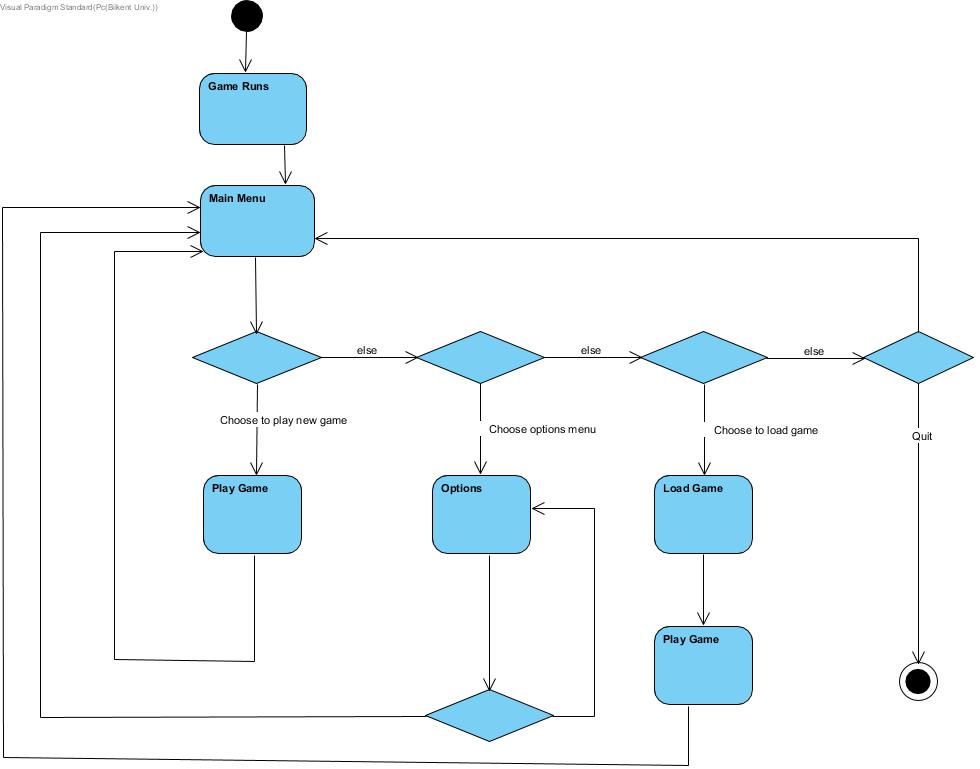
Figure 10: State Diagram of the Time Mod

The time mod is almost identical to the Classical Mod, except the game has a time limit. The steady state can direct to another final state which implies the time has run out.

* + - 1. **Golden Territory State Diagram**

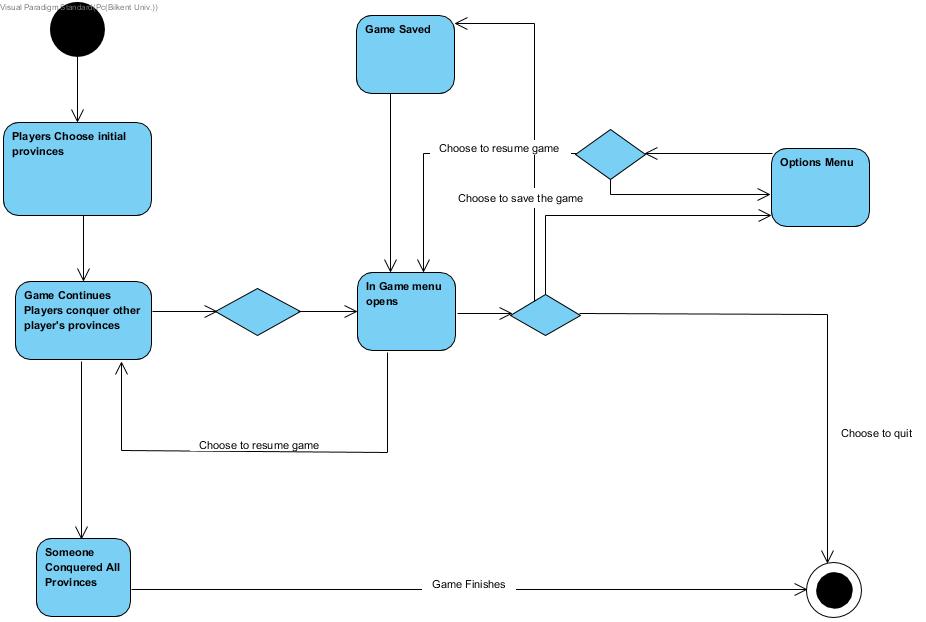
The golden territory mod’s state diagram is identical to the classical mod’s state diagram except the final state occurs when one player occupies the golden territories.

* + 1. **Activity Diagrams**
       1. 5.2.3.1 Main Function Activity Diagram



# When we run the game the Main Menu opens. There are 4 buttons which are Play Game, Options, Load Game and Quit. If you choose Play Game, a new game will be created. Else if you choose Load Game, you will resume your older games. Else if you choose Options, Options Menu will open. If you choose quit, game will close.

# Play Game Activity Diagram



# When a new game starts, first players need to choose their initial provinces. After that players start to move their troops around the world and conquer other player’s provinces. The game will end if any player conquers all needed provinces . If one of the players opens ın game menu, he/she can choose to save the game. Otherwise, he/she can choose to open options menu. Else player can choose to quit game.

* + 1. **Object and Class Models**

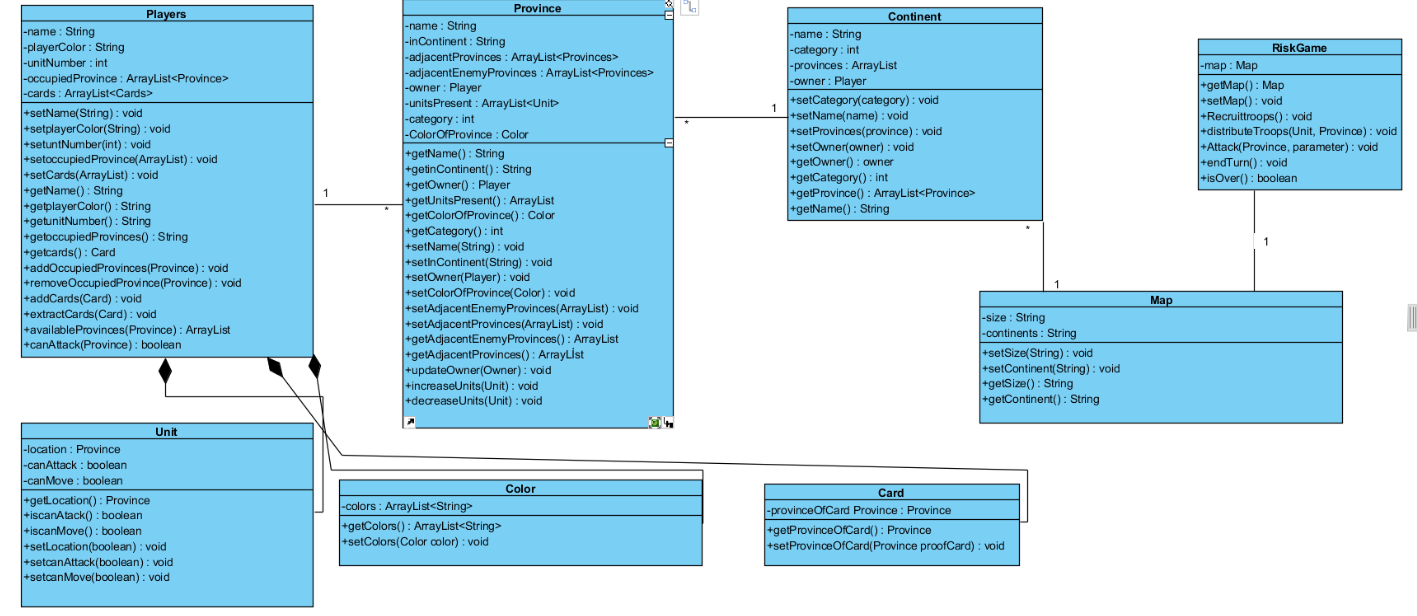


Figure 11: Class Diagram of the Game of "Risk"

Players Class:

This class defines a simple player.unitNumber holds the total number of units that the player holds. OccupiedProvinces holds the current territories of the player and the cards hold the cards that the player has in his hands. addOccupiedProvinces(Province) increases the OccupiedProvinces arraylist by the newly added Territory which is in the province parameter. removeOccupiedPorvinces is the same equivalent except it is fort he times when player loses a territory. availableProvinces(Province) show whichProvinces are available for attack. canAttack specifically shows whether a certain territory is available for attack.

Unit Class:

This class is a simple unit. Location declares which territory that the unit is currently in.setLoaction(Province) is for changing the location of the units. Unit class is a composition class, its existence is entirely dependent on the Player class.

Color Class:

By using this class , we may assign color to province and players.

Card Class:

Defining cards that players need.

Province Class

Defining a simple province and assign it to continent. This class also updates owner and units of itself each turn. The inContinent shows which continent that the Province is in. adjacentProvinces are the neighbour territories to the Province. adjacentEnemyProvinces are the close enemy territories to the Province. Owner shows the current player that has the province. UnitsPresent are the total units that are present in the Province. updateOwner() changes the current owner of the province. increaseUnits() adds additional units to the province

Continent Class

Defines continent. Provinces shows which provinces are in the Continent. setOwner(Player) is for adjusting an owner to the continent whenever a continent is occupied.( All provinces are occupied).

Map Class:

Defines a map that is needed for a game.

RiskGame Class:

In this class , game is executed. Map holds the map that the game is being played on. recruitTroops(),distributeTroops(Unit,Province),Attack(Province),endTurn() allows the main stages of the game.

* + 1. **User Interface**



Figure 12: UI of the Main Menu

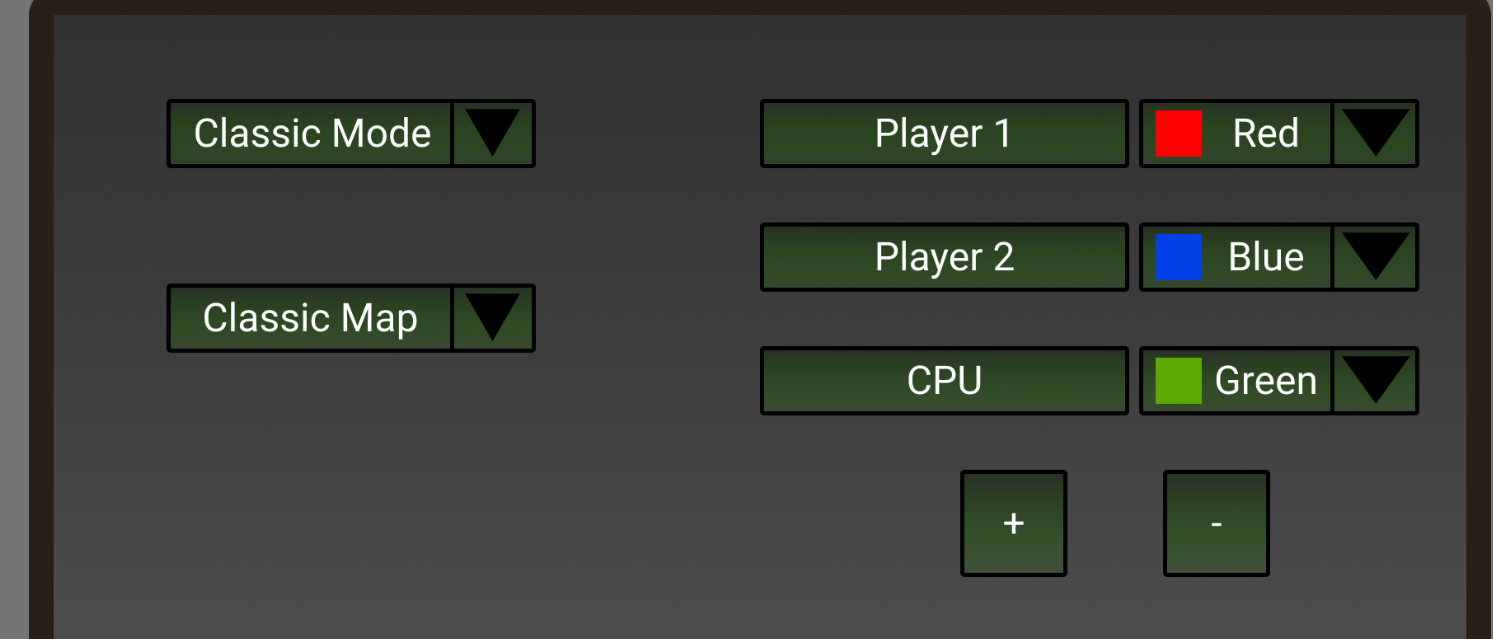


Figure 13: UI of the Pre-Game-Settings

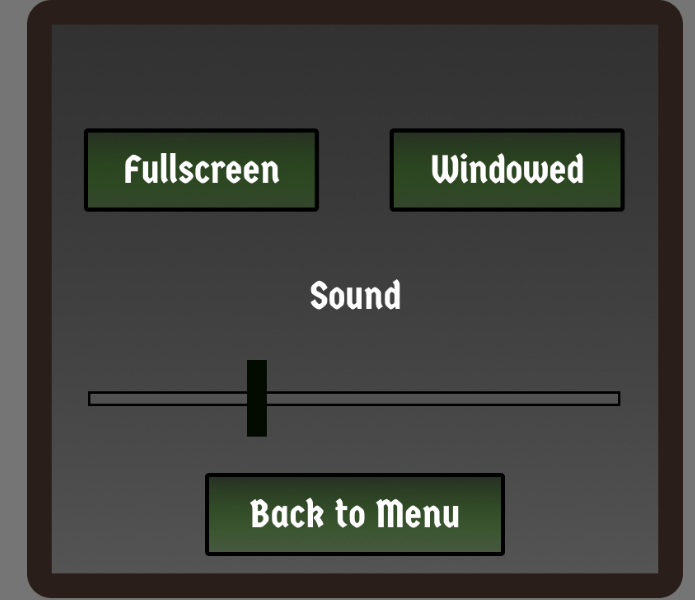


Figure 14: UI of the settings



Figure 15: UI of the gameplay

# References

[1] “Risk Online”, Pogo,[Online] Available : <https://www.pogo.com/games/risk?sl=2&gamekey=risk>

[2] “Risk”, Wikipedia,[Online] Available: <https://en.wikipedia.org/wiki/Risk_(game)>

[3]”Image of the Risk Game Map”, [Online] Available: <https://www.letsplayriskonline.com/majorcommand-com/>