****

**Object Oriented Software Engineering Project**

**Analysis Report**

**CS 319 Project: RISK: LOTR**

**Group 1J**

* **Miraç Vuslat Başaran**
* **Hazal Buruk**
* **Elena Çina**
* **Doğa Zeynep Germen**

**Intructor: Bora Güngören**

**Table of Contents**

1. **Introduction**
2. **Game Overview**
3. **Functional Requirements**
4. **Non-Functional Requirements**
5. **System Models**
6. **Use-Case Model**
7. **Introduction**

For CS319 Group Project, we are planning to create a strategy game titled “RISK: LOTR.” It is inspired by the original RISK Table Game and it is set on the imaginary map and world of “Middle Earth,” which is a part of JRR Tolkien's fantastic Lord of The Rings Universe.

The game will be a desktop application and it will be a multiplayer game. Each player will take turns on the same computer, making their moves. The game will be played with a mouse and friends.

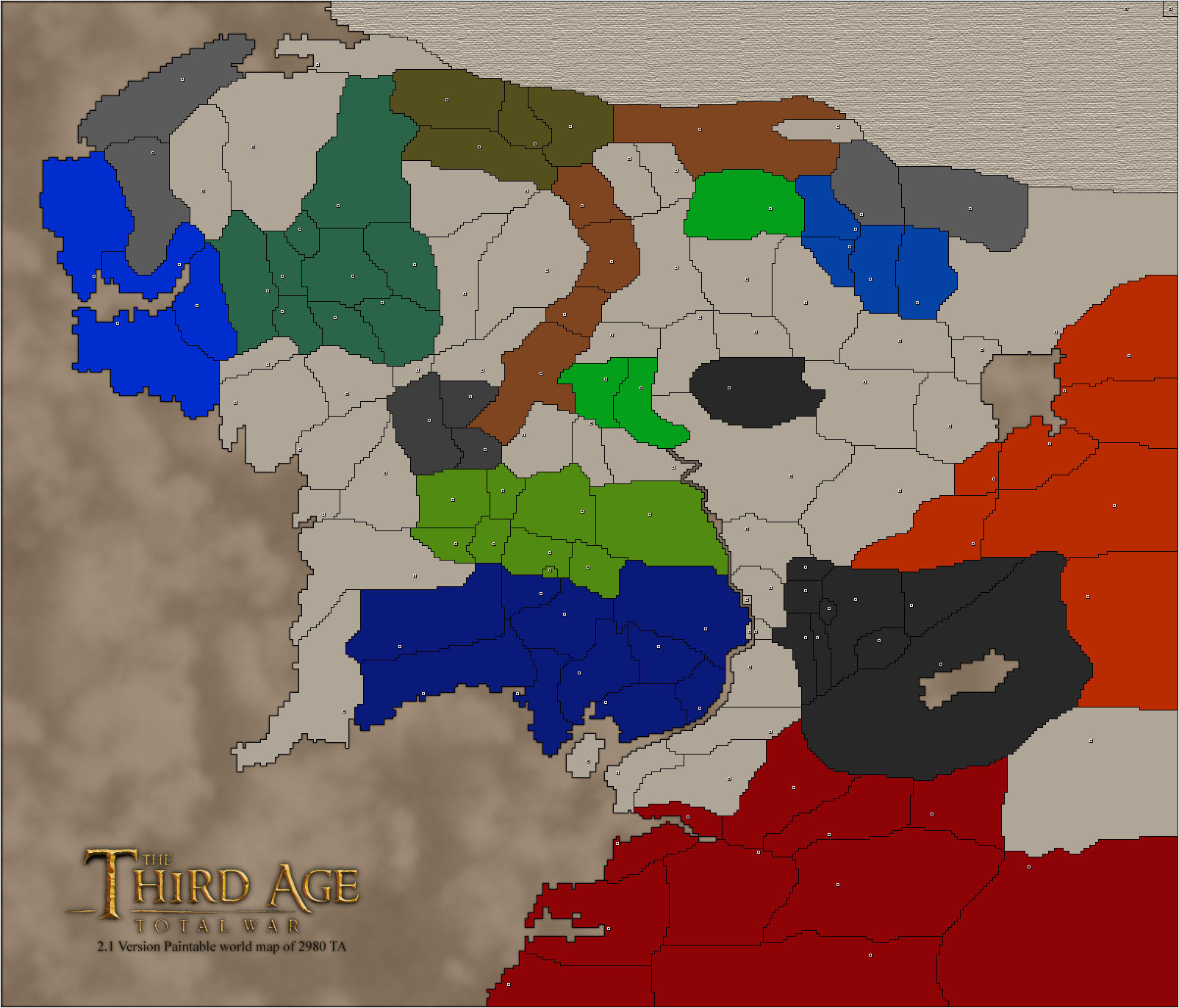
This report contains an overview of the game, explaining the overall structure of the game and the features we plan to implement; functional and non-functional requirements of the game, and finally a use-case diagram to explain the use-cases of the game.

1. **Game Overview**

RISK: LOTR is a turn-based strategy game set on the imaginary map and world of “Middle Earth,” which is a part of JRR Tolkien's fantastic Lord of The Rings Universe. The game map is divided into areas, and areas in turn are divided into provinces.

The game is multiplayer-only and each player plays on the same computer, taking turns. Each player will be assigned some provinces at the beginning of the game. The aim of each player is to conquer others’ provinces and areas to finally own all of the provinces in the map, in which case she will have won the game. A player loses the game if she has no provinces left. So, the game continues until there is only one player left and the last remaining player is the winner.

**2.1) Map, Areas and Provinces**

The game map is based on the map of the Middle Earth, and particularly from a map of a Medieval II: Total War mod named “Third Age Total War.” It can be seen below: 

As can be seen on the map, it is divided into enclosed regions of land which will be referred to as provinces. Each province has one or more adjacent provinces, so all of the provinces in the game are connected. Each province has an owner and the owner of a particular province may, and most certainly *will*, change during the course of the game.

Each province provides an income (*gold*) to its owner at the start of each turn. The players use their income to produce soldiers. So, controlling provinces are vital to winning the game.

The player deploys the armies she produces on a province. The armies that reside in a particular province may be *move*d to adjacent provinces or may be used *attack* adjacent provinces in the course of a players turn.

Besides provinces, there are *areas* in the game. Areas are composed of certain provinces in the game. For example, there are several areas such as Gondor, Mordor, Rohan etc. Mordor, for example is the total region spanned by the black painted provinces in the map.

An area provides unique bonuses *if* a player controls all the provinces that belong to that area. For example, the most basic and also important bonus an area provides is extra income. Control of areas is extremely important in the game as someone with more provinces but less areas might have less income than someone with less provinces but more areas.

**2.2) Army Types**

There are many unit types in the game. Some of these can be produced by all factions while some others are faction-exclusive. Each army type has different attack and defense values and different vulnerabilities. Also, each army type has one or more potential tactical advantage modifier related to it. Army types in the game are as follows:

**Swordsmen**

This is most basic infantry unit in the game. It is an army composed of infantry holding swords. Its attack and defense values are moderate and it has a high vulnerability. Swordsmen generally serve as the bulk of the army.

**Axemen**

Axemen are the second type of infantry. They are equipped with axes and have higher attack and lower defense capabilities than swordsmen. They also have high vulnerability.

**Spearmen**

Spearmen are the third type of infantry. They are equipped with spears and have both lower attack and defense capabilities than swordsmen. So, what is their advantage over swordsmen? Spearmen are *very* effective against cavalry which makes them an extremely important unit in certain cases. They also have high vulnerability.

**Archers**

Archers are a missile unit. They are equipped with bows and have lower attack and higher defense capabilities than swordsmen. Thus, they are effective in defending a province and could potentially serve as garrison units. They also have high vulnerability.

**Cavalry**

Cavalry are a mounted unit. They have higher attack and defense capabilities than infantry and are particularly effective against swordsmen, axemen and archers. However, they are *very* weak against spearmen. They are more costly to produce than infantry but have lower vulnerability.

**Warg-Riders**

Warg-Riders are exclusive to Isengard and serve as a replacement for cavalry. Although their capabilities are similar, warg-riders are better at attack than defense compared to cavalry. They cost the same as cavalry and have the same vulnerability.

**Mumakil**

Mumakil are exclusive to Harad and are one of the most powerful units in the game. They have higher attack and defense capabilities than cavalry and warg-riders but consequently are more costly. They have very low vulnerability and are hard to destroy.

**Nazgûl**

Nazgul is arguably the strongest unit in the game. They can only be produced by Mordor and have the best attack and defence capabilities in the game. However, they are the costliest unit in the game but have the lowest vulnerability.

**2.3) Battle System**

The battle system is the central aspect of the game. In the traditional RISK Board game, there is a very simple and un-realistic battle mechanism. In contrast, RISK: LOTR has a battle system that portrays the battles that takes place in the Lord of The Rings Universe in a very realistic way. It takes into account many inputs and thus is a simple but realistic simulation.

Each battle has two sides: An attacker and a defender. The attacker is the one attacking to the defender’s province from an adjacent province. The attacker decides how many units to send from the attacking province to the defending province.

The armies of both sides meet each other at the battlefield. However, each battle has a maximum combat width, which is by default 5. Combat width is the maximum amount of units one side can field in the battle in a phase of battle. This is to portray the fact that you cannot attack with all your forces at once. At the beginning of the battle, each side fields the total amount of units it has if it is less than maximum, or field exactly combat width amount of units (5) in the battlefield. The rest of her armies are reserves and may come into play in the progress of a battle.

Each battle is composed of phases. In the beginning of each phase, if a side has less number of units than maximum combat width and has reserves, units from the reserves are randomly chosen to fill the battlefield until the maximum combat width is reached.

In the second phase, the actual “fighting” begins. As explained previously, each unit has unique attack and defense values. These values are of the following form: “*xdy+z.*” The notation xdy+z means throw a y-sided die x times and add z to the sum of these dice. So, for example a swordsmen has 1d5 attack and 1d5 defense whereas an axemen has 1d5+1 attack and 1d5-1 defense. In this phase, for each unit in the attacker side, the *“result”* of the units attack is calculated by “*throwing a dice*” (which is basically creating a random number with the given constraints). Then, the sum of the “*attacks*” of these units is taken. And finally this sum is multiplied by *tactical advantage modifier*. Tactical advantage modifier is “1.00” by default. However, the composition of both the attacker’s and defenders armies in the battlefield, the place where the battle takes place and some faction-related modifiers may increase or decrease the *tactical advantage modifier*. For example, if the attacker has no infantry whatsoever in the battlefield, she will incur a tactical advantage penalty or if she has spearmen and the defender has cavalry, she will receive a *tactical advantage* bonus.

The same procedure is applied for the defender’s case. This time, however, defender’s units’ defense values are taken into account rather than attack values. Finally, the result of both parties’ attack and defense values are compared and the one who has a higher value wins that phase of battle.

In the next and final phase of a battle, the defender, because she lost this particular stage of battle, loses one of her units. As mentioned previously, each unit has a vulnerability attribute. A unit amongst loser’s armies on the battlefield is chosen randomly to be destroyed. However, this selection is not totally random and is biased to destroy units with higher vulnerability than units with lower vulnerability. So, your chances of losing your hard-earned Nazgûl is lower compared to losing a common, cheap swordsmen.

After loser side loses one of her units, if she still has at least one soldier on the battlefield, the battle continues to first phase of next stage. So, this loop continues until a side loses all her armies. When one side loses all her armies, she loses the battle. If the attacker wins, she gains the province the battle takes place on. If the defender wins, she gets to keep her province.

**2.4) Factions**

At the start of the game, each player chooses from a collection of corresponding to a race or a kingdom from Lord of The Rings Universe. Each faction has different qualities in terms of armies it can yield and the strength of these armies. Also, some factions have gameplay characteristics that set them apart from others. The factions in the game are:

**Rohan**

Rohan is the one of the two kingdoms of men in the Middle Earth. They are most notably known for their cavalry and their horde-like lifestyle. In the game, a Rohan player can yield all the basic army units. However, one might be inclined towards producing mostly cavalry as Rohirrim (people of Rohan) Cavalry1 have a huge tactical advantage over other factions’ cavalry and also other units.

**Gondor**

Gondor is the other kingdom of men in the Middle Earth. It is historically *thekingdom of men*. The people of Gondor value honor and tradition and they are willing to protect their land to the last man. That is why a Gondor player has some tactical advantages when fighting in the Area of Gondor. A Gondor player can also yield all the basic army units.

**Elves**

Elves are the immortal, pure beings of the Middle Earth. They are very elegant and timeless. They can produce all the basic army unit types. However, they are most notably known for their archery skills. This is why Elvish archers have an advantage over other factions’ archers.

**Dwarfs**

Dwarves are a short but muscular and sturdy race of the Middle Earth. Albeit small, they are very strong. That is why, their axemen are the best axemen in the game. Unfortunately, since they are small, they cannot ride horses and thus cannot produce cavalry.

**Mordor**

Mordor is the land to the east of the Middle Earth and it is where Sauron resides. Mordor is the most evil faction of the game. It is composed of mostly orcs and that is why a Mordor player cannot yield cavalry units, which is a big disadvantage. However, they make up for this disadvantage by their ability to produce Nazgûl, which is arguably the most powerful unit in the game.

**Harad**

Harad is the second evil faction of the game. They, unlike Mordor, can produce all the basic army units including cavalry. They also can produce “Mumakil,” which are gigantic war elephants.

**Isengard**

Isengard is the third evil faction of the game. Isengard hosts Uruk-Hai, which is basically a stronger version of orcs. They cannot produce cavalry much like Mordor but they can produce Warg-Riders to make up for this.

**2.5) Gameplay**

The game starts with choosing how many players will play the game. After the number of players is chosen, the game map and the GUI appear on the screen. Each side starts with a fixed number of soldiers. Inıtially, the game map is empty, meaning that none of the provinces is occupied. Each player takes turns deploying a single army unit on an unoccupied province or their own province on the map. When they deploy a unit to an unoccupied province, they from then on occupy that province. This process lasts until everyone places all their undeployed armies.

After the initial occupation/deployment phase at the start, the game progresses in a linear manner with each player taking turns playing out their moves. Each player’s turn has two phases:

The first phase is the **deployment phase**. In this phase, the player receives her income from the provinces and areas she controls. With this income, she can produce units and deploy them on provinces she controls. She ends this phase by clicking the *“End Deployment”* button.

The second phase is the **mobility phase**. In this phase, the player can transfer her units in a province to the adjacent provinces. Alternatively, she can attack from a province to another player’s adjacent province. In both attacking and transferring, she decides how many of each unit type she wishes to transfer or attack with. She ends this phase by clicking *“Execute.”*

After each player plays out their two phases, the turn’s final phase, which is the **execution phase** play out. In this phase, firstly, each players’ deployment orders are executed. Then, their transfer and attack orders are executed and the result of any battle is displayed on the screen. If there are any territorial changes, these changes occur after each battle. After this phase, a check is made to see if a player has lost all her provinces. If she does, it means she lost the game. If there is more than one player in the game, the game is continued to the next turn. If there is only one player left, she wins the game and the game finished.

**3) Functional Requirements**

**3.1 Choose Players**

Since this game is multiplayer and at the beginning of the game the number of players that are going to play the game should be specified.

**3.2 Start Playing**

After the game is open through this button, the first player is able to start playing.

**3.3 View Help**

This option helps players to understand the game by providing an explanation of the logic of the game, strategies that should be followed and some instructions. Also the personal and contact information of the developers of the game is provided here.

**3.4 Restart**

At any point during the game the player can decide to restart the game from the beginning with the default properties set.

**3.5 Adjust Sounds**

Through this option users are able to set or mute the background music. Also it allows the user to turn on additional specific sounds. For instance, a victory sound could be played when a faction wins a battle.

**3.6 Check Gold**

Each player can check the amount of gold he has collected from the provinces he has conquered.

**3.7 Produce Soldiers**

The player can check the value of each soldier and based on the amount of gold that has been collected she can produce soldiers.

**3.8 End Deployment**

After a player performs her deployment, this option allows her to save the movement she performed and switch to the mobility phase.

**3.9 Simulate Battle**

In accordance with the attacks and transfers that the player performs this function simulates the corresponding battle. It will be called several times in the execution phase where the outcomes of the battles will be displayed.

**3.10 Execute**

After each player performs her deployment, attack and transfer actions; clicking this option will switch the gameplay into execution phase. In this phase, any possible battles and hence the winners of those particular battles will be displayed.

**4) Non-Functional Requirements**

**4.1 User-friendly interface**

This game will provide a user friendly interface for the players. The areas owned by a particular faction will be clearly shown and pictures corresponding to each unit will be provided in order to make it easier for the players to follow the game.

**4.2 Game Performance**

This game consists of a lot of “actions” which will be performed by the players through the mouse, so it needs to have a low time delay. Also, there will be a lot of simulations and calculations in the execution phase and they need to be executed in such a small time that the user is not affected by the computation overhead.

**4.3 Logic of the Game**

An important requirement is that understanding and playing the game needs to be easy. Having multiple types of soldiers and factions, also the system of “gold” will be simple to understand so that every player will be able to get along quite well with this game.

**5) System Models**

**5.1 Use-Case Model**

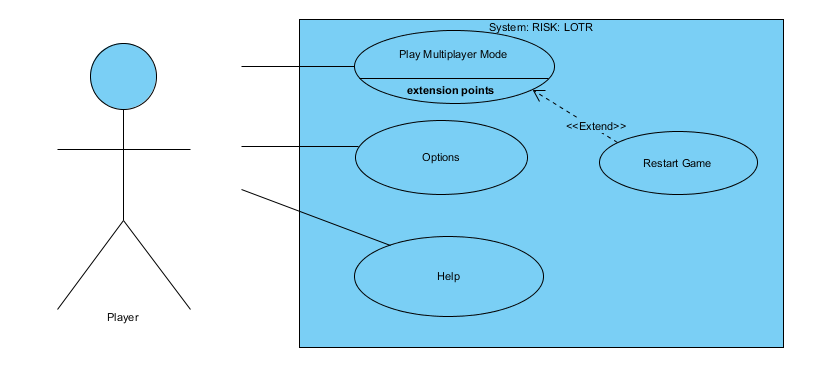


Figure 1: Use Case Diagram of the game

**Use Case 1:** Play Multiplayer Mode

**Actors:** More than one player on the same computer

**Entry condition:** Player has opened the game.

**Exit condition:**

* One player beats all of the other players, OR
* One of the players click “Exit Game” button.

**Main Flow:**

1. Game starts
2. Player selects the “play multiplayer mode”
3. Game asks the number of players (min 2 - max 7)
4. Player chooses the number of players.
5. Game asks the name of the player and the factions to each player.
6. Each player chooses one faction.
7. Players try to conquer all the regions on the map.
8. A player conquers all the area on the map.
9. Game finishes.

**Alternative Flow:**

-One of the players chooses to exit the game

**Alternative Flow 2:**

-One of the players chooses to restart the game. Game turns back to the beginning of the 7th step.

***Use Case 2:*** Options

**Actors:** One of the players

**Entry condition:** Game has already opened and one of the players presses the 'Options' button.

**Exit condition:** One of the players click 'Exit Options' button.

**Main Flow:**

1- One of the players presses the 'Options' button.

2- Sound options appear on the screen.

3- Player sets the sound effects and music options arbitrarily.

4- Player presses the 'Exit Options' button.

***Use Case 3:*** Help

**Actors:** One of the players

**Entry condition:** Game has already opened and one of the players press the 'Help' button.

**Exit condition:** One of the players click 'Exit Help' button.

**Main Flow:**

1- One of the players presses the 'Help' button.

2- Possible problem titles are given as separated sections.

3- Player chooses the section which is related to problem.

4- Under the selected section, information and solutions are given.

5- Player presses the 'Exit Help' button.

***Use Case 4:*** Restart

**Actors:** One of the players

**Entry condition:** Game has already opened and one of the players press the ‘Restart' button.

**Exit condition:**

* One player beats all of the other players, OR
* One of the players click “Exit Game” button.

**Main Flow:**

1. One of the players presses the 'Restart' button.
2. The game starts from the beginning with the specified factions for every user. But all the soldiers and the region changes are resetted.
3. Players try to conquer all the regions on the map.
4. A player conquers all the area on the map.
5. Game finishes.

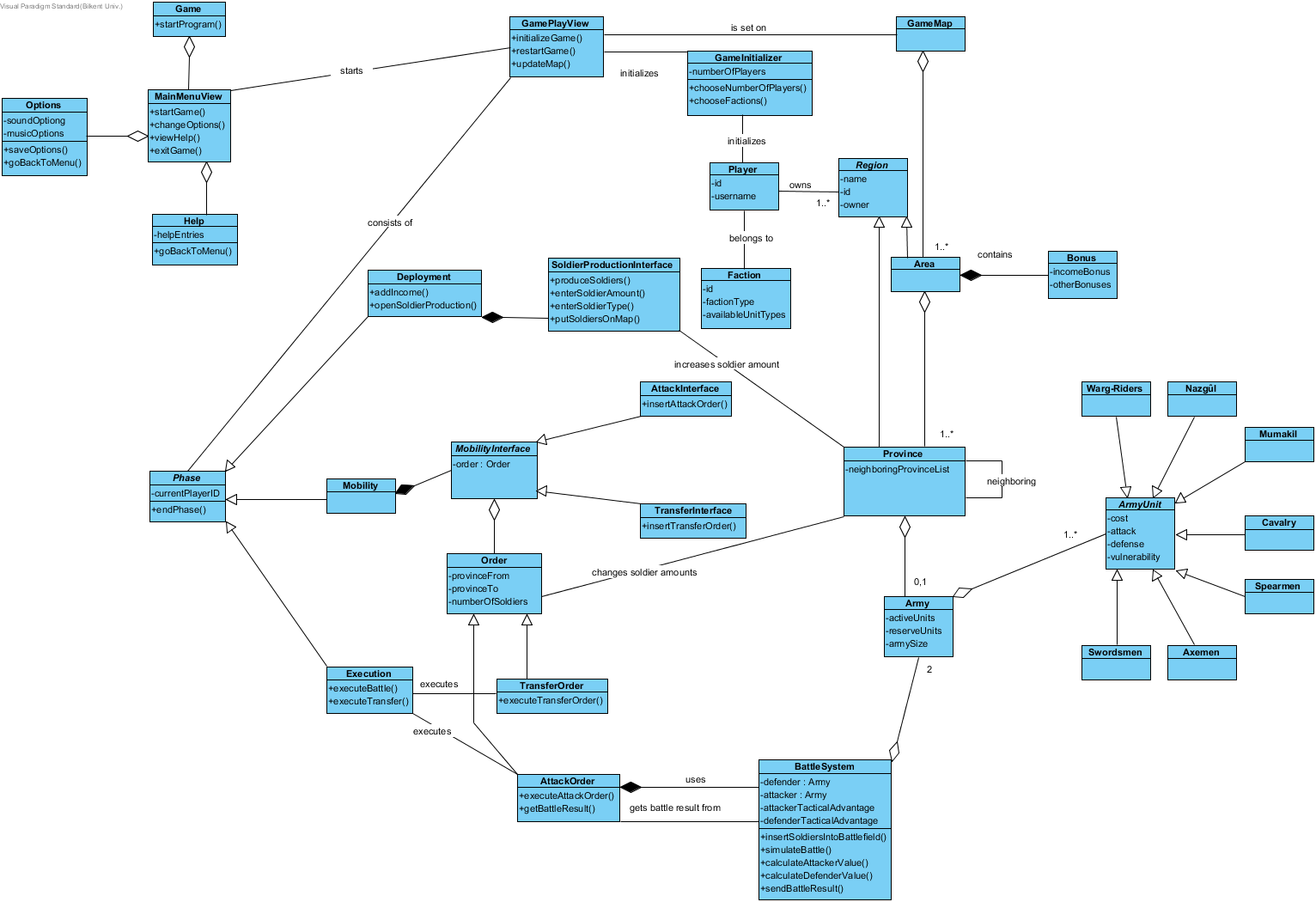
**Alternative Flow:**

-One of the players chooses to exit the game

**Alternative Flow 2:**

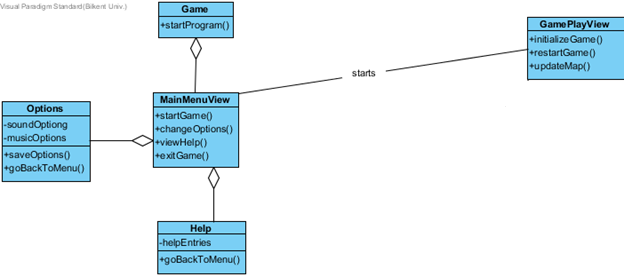
-One of the players chooses to restart the game. Game turns back to the beginning of the 2th step.

* 1. **Object and Class Model**



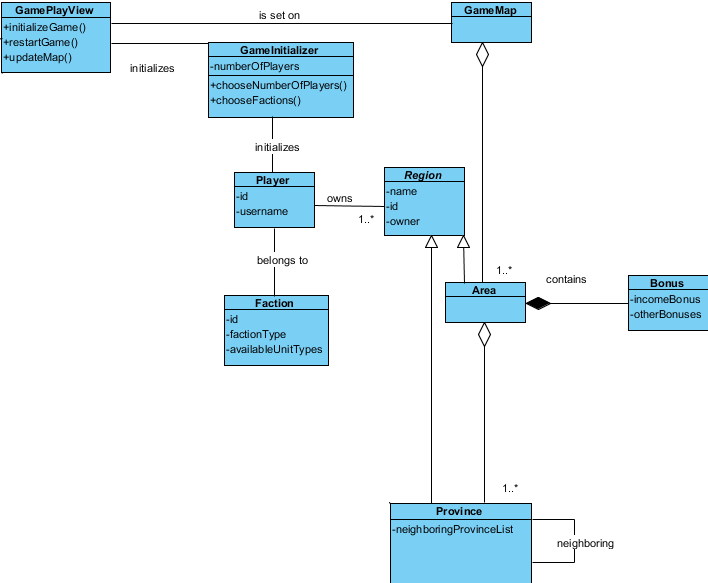
Class diagram of RISK: LOTR is displayed above entirely. Because there are many classes, it may seem complex and hard to understand. That is why, we divided into parts in the following subsections.

* + 1. **Main Menu**



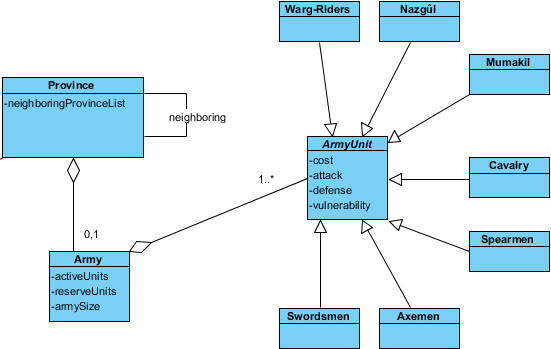
Above, one can see classes related to Main Menu. The game starts from the main menu and the user can access Options and Help from here. Also, user can start the game and switch to GamePlayView.

* + 1. **GamePlay View**

****

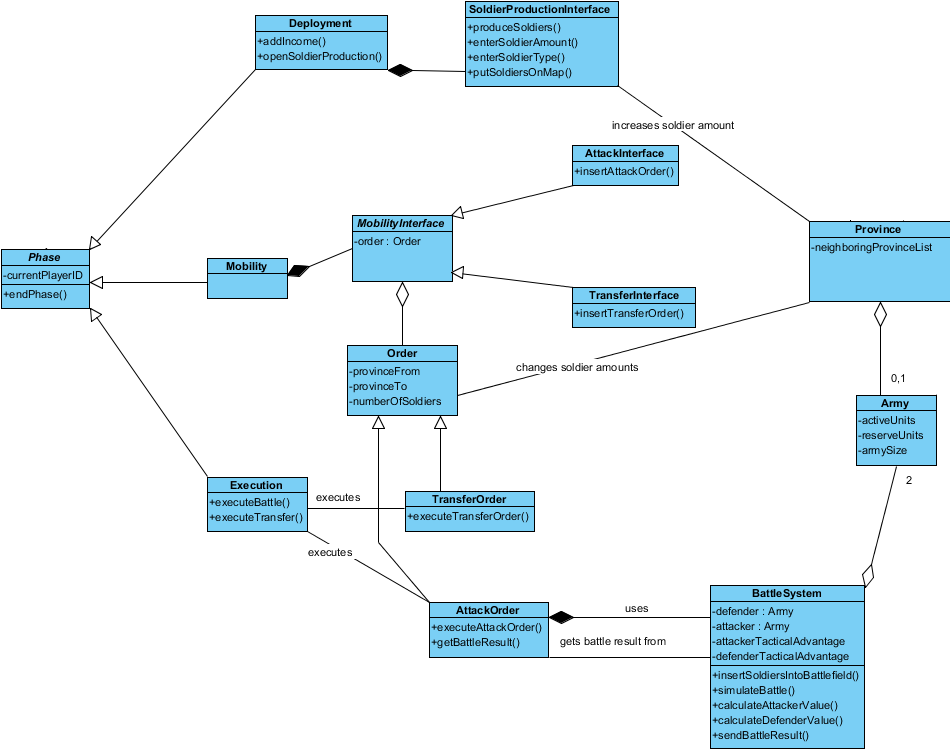
These are classes related to the GamePlayView where the game actually takes place. We have the components of the map, the map itself; players and corresponding factions and the GameInitializer Class that sets up the game on the first turn.

* + 1. **Armies**

****

Above are the Army-Related classes. Provinces may have an army on them. The Armies consist of ArmyUnits which could be one of the several types of Units specified such as Cavalry or Spearmen.

* + 1. **Game Phases**



A GamePlay consists of Phases. There are 3 different kinds of Phases: Deployment, Mobility and Execution. Each Phase has its own unique interface with which the players play out that phase. In the Deployment Phase, there is SoldierProductionInterface for the production of soldiers. In the Mobility Phase, there are AttackInterface and TranferInterface to either attack or transfer soldiers. Finally, in the Execution Phase, there are no interfaces but there are Order child classes TransferOrder and AttackOrder that carries out the orders given in the Mobility Phase. AttackOrder uses the complex BattleSystem Class to simulate the battle and get the result.