



Bilkent University

Department of Computer Engineering

# CS319 Project - Risk

## Analysis Report

### **Group Members:**

Adeem Adil Khatri	21801174
Abdulmalak Albeik	21801277
Mannan Abdul	21801066
Maryam Shahid	21801344
Osama Tanveer	21801147
Yahya Mahmoud Ahmed Elnouby Mohamed	21801332

Group: 1-G

Section: 01

Instructor: Eray Tüzün

Teaching Assistants: Barış Ardış, Emre Sülün, Elgun Jabrayilzade

# **Contents**

1. Introduction
2. Overview
  - 2.1 Gameplay
  - 2.2 Risk play pieces and their use
3. Functional requirements
4. Nonfunctional requirements
5. System models
  - 5.1. Use case model
  - 5.2. Dynamic models
  - 5.3. Object and class model
  - 5.4. User interface
6. Glossary and references

# **1. Introduction**

Risk is a strategic board game that represents diplomacy, conflict, and conquest that is played by two to six players. The game is played on a map that contains 6 continents that include 42 territories, with real-life geographical borders and political regions [1]. The game was first introduced in France in 1957, with the name *La Conquête du Monde* (The Conquest of the World), which was invented by Albert Lamorisse, a French film director [1]. In 1959, the game's rules were modified by the Parker Brothers and were released as *Risk: The Continental Game*, then as *Risk: The Game of Global Domination* [1]. The board game is composed of the map, the army troops, including infantry, cavalry, artillery pieces, 5 - 6 dice, and risk cards, that are composed of cards that represent each territory, mission cards, and wild cards. Players play in turns, using the army pieces to try to conquer other player's territories. The main aim of the player is to attack and occupy other player's territories and eliminate them from the game. The last standing player, who conquered all territories, is the winner of the game.

# **2. Overview**

The Risk game we are going to implement will be an online version of the classical board game, that can be played by 3 - 5 players and on different screen sizes. The general rules of the board game will be implemented. Our version of the game is going to be implemented using React.js and Redux for the frontend, Django for the backend, and MySQL relational database for storing the game's data including players' scores and information, and saved games to be able to load them in the future again. The game will contain different maps that players can select between, including the original map of the risk board game. The players will have the option to save and load games if they want to stop the game and continue it later. Besides, high scores and previous players' scores will be tracked and saved.

## **2.1 Gameplay**

The gameplay will be very simple for the players to use. The main player at first would log in if he has an account or sign up if he does not have an account. After the login is successful, the player will be asked to enter the number of players, which can be 3 – 5 players inclusive. After the number of players has been selected, each player will start by

rolling only one dice, which can be done by pressing the roll button that appears for each player. The player that receives the highest dice number will start by deploying one troop in a country of his/her choice, then moving counterclockwise the players will deploy one troop each until all the countries on the map are occupied. To deploy troops the players, in turn, would click on the country that they want to occupy. After all the countries are occupied, the number of troops for each player will be calculated. Starting with the first player, who was decided earlier by the highest dice roll, and moving counterclockwise, the players, in turn, will start by deploying the troops they received in the countries that they occupy after the player finishes deploying his/her troops, the player would have two buttons to choose between, either to pass or to attack. If the player decides to pass, the next player repeats the process starting from deployment and then choosing either to attack or pass.

On the other hand, if the player decides to attack a country, he/she can only attack the bordering countries to his/her country that he is attacking from and must leave at least one troop behind in the country that the attack is coming from. For the player to attack a neighboring country, he/she will press on the country they want to attack. The player attacking and the player being attacked both receive dice. Both players will roll the dice by pressing on the roll button and the highest dice from each player will be compared, and accordingly the winning and losing player will be decided. If the attacking player wins, he/she must move troops to the country conquered and receive a risk card. The player can decide to attack again or to pass. After collecting the number of cards required, the player may trade them in for additional troops by clicking on the trade button. In the end, the winner is the one who conquered all countries on the map. The players can stop playing at any time, saving the game's progress and loading it later when they want to continue playing.

## **2.2 Risk Play Pieces and their use**

This version of the game will include 3 different maps, including the original map of the board version. There will also be 5 different colors for the armies to be chosen from. The army pieces will be similar to the board game, in which there will be 3 different pieces. The army pieces include the infantry, the cavalry, which is counted as 5 infantry, and the artillery that is counted as 10 infantry or 2 cavalries. Besides, the game will include risk cards that will only contain the territory cards. And Finally, 5 – 6 dice that will include 2 colors 3 dice for each.

At the start of the game, the players will be able to choose between different map versions to play on. The army pieces will be used for attacking and conquering the countries. The number of infantries at the start per player will be decided using the formula  $50 - 5n$ , where  $n$  is a number between 3 and 5 inclusive. Later in each turn, the players will receive  $x / 3$ , disregarding the remainder, pieces of troops, where  $x$  is the number of countries owned by the player. The players can receive additional troops by trading in cards they have. The players will deploy the troops they received, and they can decide either to attack or pass. In attack, the attacking player receives 3 dice of one color and the defending player receives two dice of the other color. The players roll the dice and the lowest dice roll by the attacking player is set aside and the highest dice roll of the attacking player is matched with the highest dice roll of the defending player and the second dice is matched with the second die. If the dice of the attacking player is higher than the defending player one of the defending player's troops is removed per dice, but if the dice of the defending player is higher than or equal to the dice of the attacking player, the attacking player loses one troop per dice.

### **3. Functional Requirements**

#### **1. Log in or sign up**

At the beginning of the game, the players will either login if they already have an account or sign up if they do not have an account

#### **2. Choose number of players to play the game**

The user would be able to choose the number of players to play the game

#### **3. Choose between different armies.**

The player can choose between different colors that will be set as the army pieces' color.

#### **4. Choose different maps**

There will be different maps including the original map of the classical board game that the players can choose between.

## **5. Deploy armies in the conquered countries.**

At the start of the game each of the players will have a certain number of troops, and each player, in turns, will start to deploy one troop per turn. Besides, in the turn of each player, the player would be able to add new pieces to the countries he/she conquered, the number of pieces that the player receives to deploy will be decided according to the game rules.

## **6. Attack neighbouring countries or pass**

After the player has deployed the armies in the countries, he/she can either attack other neighboring countries or skip attacking and moving on to the next player.

## **7. Win a new card for every new country conquered**

If the player succeeded in conquering a new neighboring country, he will receive a card for every new country he/she conquered, after collecting a certain number of cards the player can substitute the cards with a certain number of army pieces to deploy in his/her countries.

## **8. Trade cards after having a certain number of cards**

After the player receives a certain number of cards as defined by the rules of the game, the player can trade them in for new troops to deploy in his/her countries. In addition, the number of troops received depends on the number of trades the player has done until he/she reaches a certain number of trades in which the number of received troops will be constant starting from that point and forward on.

## **9. Save and load the game**

The players would be able to save the game to the point they stopped at and can later on load it again.

# **4. Non Functional Requirements**

## **1. Performance**

The game will be built to work smoothly for a user. At any time, the frame rate will not drop below 30 frames per second so as to be noticed by the players.

## **2. Accessibility**

The screen at all times will have no more than 5 buttons to ensure that the player is able to easily access them. The icons for the buttons will be used purposefully to allow the user to interact with them easily.

## **3. Experience**

The player who is going to take a turn will be presented with guidance arrows and popups to guide them towards possible moves. This will increase the user experience.

## **4. Maintainability**

The codebase will be written to ensure that in the future if the number of players who can play the game is to be updated, it can be done with the minimest possible changes. Reusable components will be used to make sure that no duplication of functionality occurs to prevent any future corruptions.

## 5. System Models

### 5.1 Use case Model

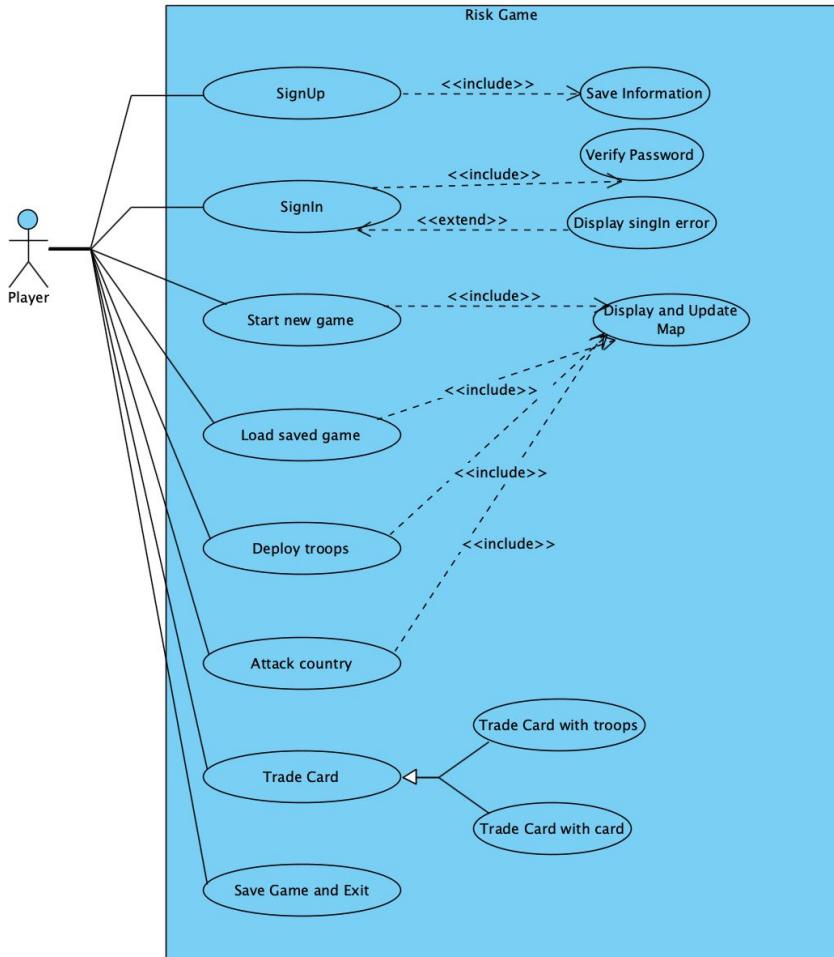


Figure 1 (Use Case Diagram)

## 5.2 Dynamic Models

### 5.2.1 Sequence Diagrams

#### Configure Game Diagram

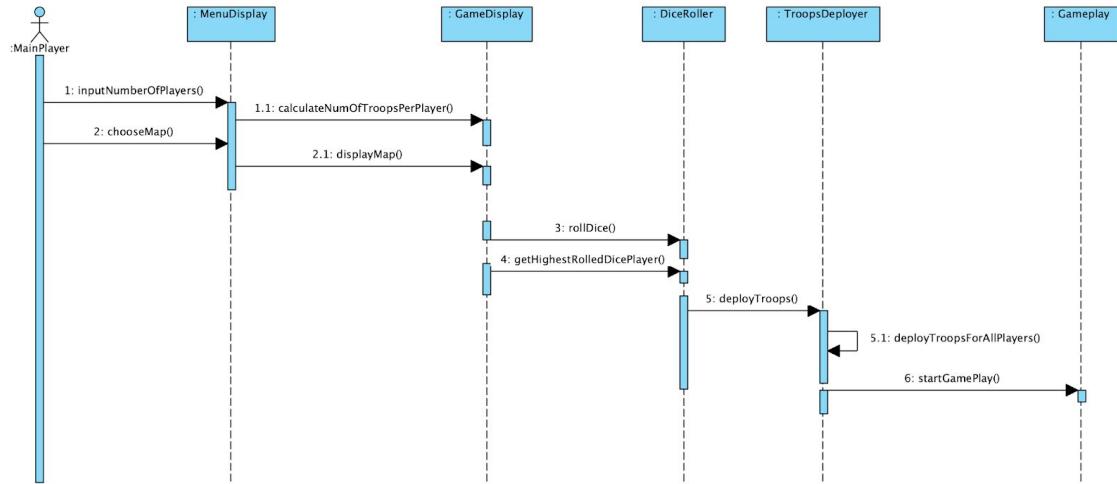


Figure 2 (Configure Game Sequence Diagram)

#### Scenario:

The main player is setting up the game configurations. He/she chooses the number of players and the map, and a series of events is triggered which causes the main setup of the game. The players then choose the color for their troops. The dice is rolled and the first player's turn is determined. Each player is then able to deploy their troops. After the troops deployment is completed, the gameplay starts.

#### Explanation:

The main player decides the number of players in the game as well as the map for the game. Upon inputting the number of players and selecting the map, the MenuDisplay sets up the game by calculating the number of troops per player and by displaying the map, by calling calculateNumOfTroopsPerPlayer() and displayMap() in the GameDisplay object. The displayMap() method allows the players to choose their troops' colors. The GameDisplay object then triggers the rollDice() and getHighestRolledDicePlayer() in DiceRoller object to roll the dice and get the player who rolled the highest rolled dice player. The DiceRoller object then calls the troops deployer which ensures that all players have deployed the troops according to the game rules by triggering

deployTroopsForAllPlayers(). Once this is done, the TroopsDeployer object triggers the startGamePlay() to start the game.

### Player Turn Diagram

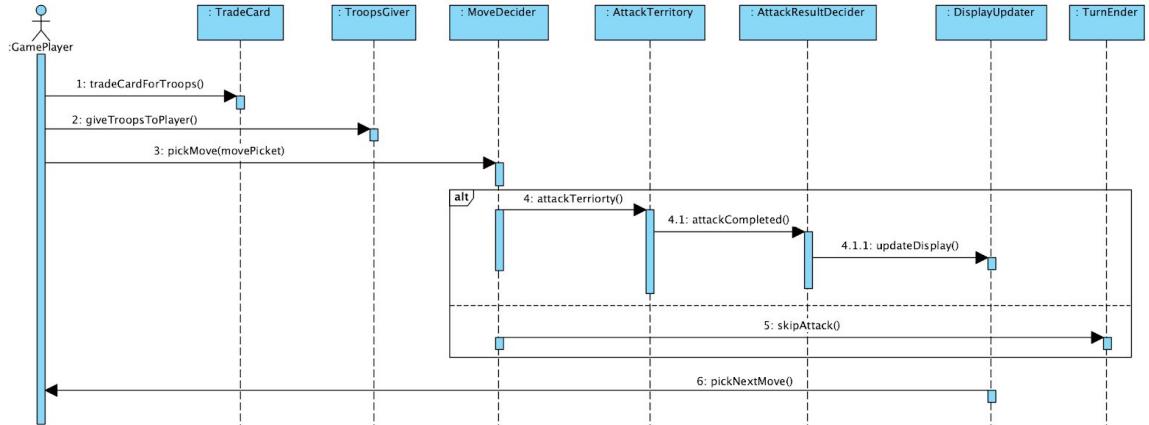


Figure 3 (Player Turn Sequence Diagram)

#### Scenario:

One player's gameplay is depicted and it is the same for every player. The player is required to trade a card according to the rules of the game. The player gets troops for their turn. The player then decides the move he/she is going to make. If the player chooses to attack another territory, the result of the attack dictates the game view. The player can then pick the next move. If the player is unable to make a next move, according to the rules of the game, the player then loses the turn and the next player takes his turn. If the player loses and is still in a position which allows him/her to attack, he/she can choose his next move. If the player wins, he/she can decide whether to attack another territory or skip his/her attack. In case the player skips his/her attack, the player's turn is ended and the next player gets the turn.

#### Explanation:

GamePlayer represents a player and triggers the tradeCardForTroops() method in the TradeCard object. GamePlayer also triggers the giveTroopsToPlayer() method in the TroopsGiver object and the number of troops as calculated by the game rules are given to the player. After this is done, the GamePlayer object triggers the pickMove() method in MoveDecider along with an argument which is an indicative of the move the player has decided to make. The MoveDecider then triggers either attackTerritory() or skipAttack(). If skipAttack() method is triggered, the TurnEnde object finishes the turn and gives the turn to the next player. If, however, attackTerritory() is triggered, a

territory is attacked and the `AttackResultDecider`'s `updateDisplay()` is triggered in the `DisplayUpdater`. The `DisplayUpdater` is responsible for updating the map, the player's troops, and is responsible to handle the player's troop maneuvering. Once this is completed, the `pickNextMove()` method is triggered which causes the player to choose the next move, which can either be skip or attack as dictated by the game rules. This cycle repeats, and upon termination, the next player has a similar cycle.

### 5.2.2 Activity Diagrams

#### Configure Game Diagram

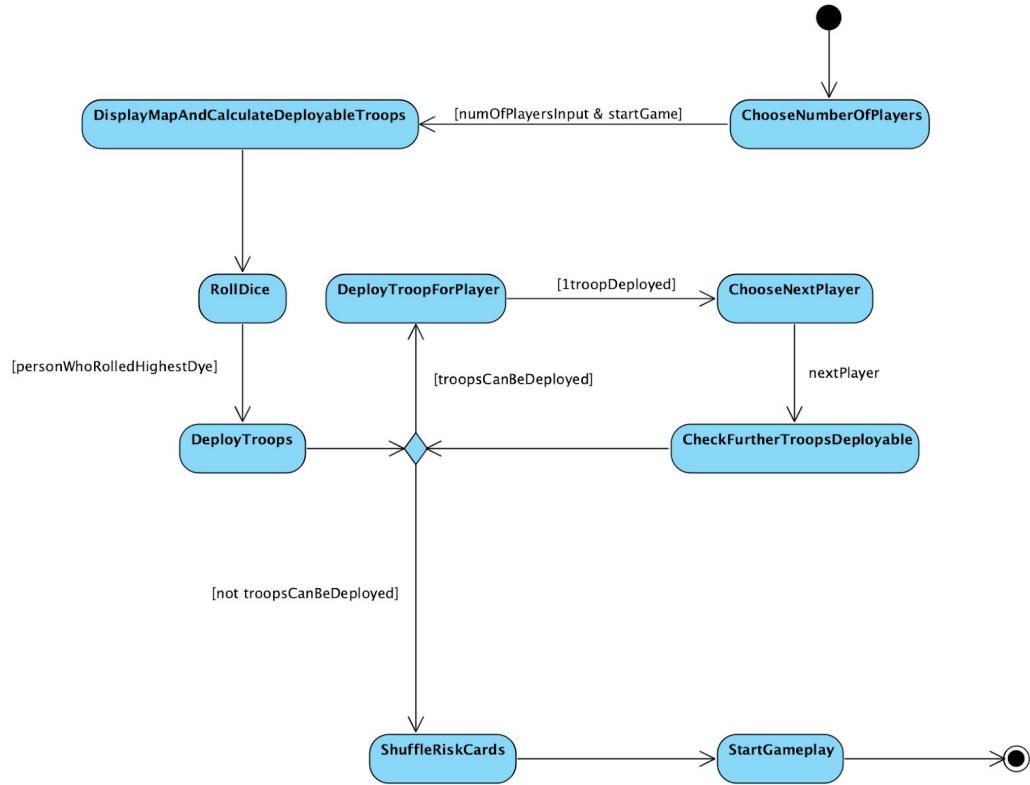


Figure 4 (Configure Game Activity Diagram)

This activity diagram represents the flow of activities during initial game setup. The main player inputs the number of players that can play the game. This number can range from 3 to 5 inclusive. Upon choosing the number of players, the main player has to start the game using a button. This will trigger the start of the activity 2. Activity `DisplayMapAndCalculateDeployableTroops` displays the map to the players and calculates the number of troops that can be deployed per person. Upon completion, a dice is rolled as indicated

by the RollDice activity, and the activity DeployTroops chooses the first player using the result of the RollDice activity to deploy troops. The rest of the players get turns in a counterclockwise manner. A decision node is reached after this activity. If troops can be deployed, we transition to the DeployTroopForPlayer activity. Initially, this will be the player who rolled the highest dice number. Since we already know by now who rolled the highest dice number, we do not need to check the player. After the player has deployed 1 troop, we transition to the activity that chooses the next players, namely ChooseNextPlayer. Upon choosing, we transition to the activity CheckFurtherTroopsDeployable. This activity checks if the player is able to deploy troops. We again reach the decision node and carry on the sequence of activities as mentioned above until troops can not be deployed. In the latter case, we transition to the activity ShuffleRiskCards. This activity shuffles the risk cards, as indicated by its name. After shuffling the Risk cards, we reach the activity StartGameplay, which triggers the start of the game. The final activity node is reached.

### Player Turn Diagram

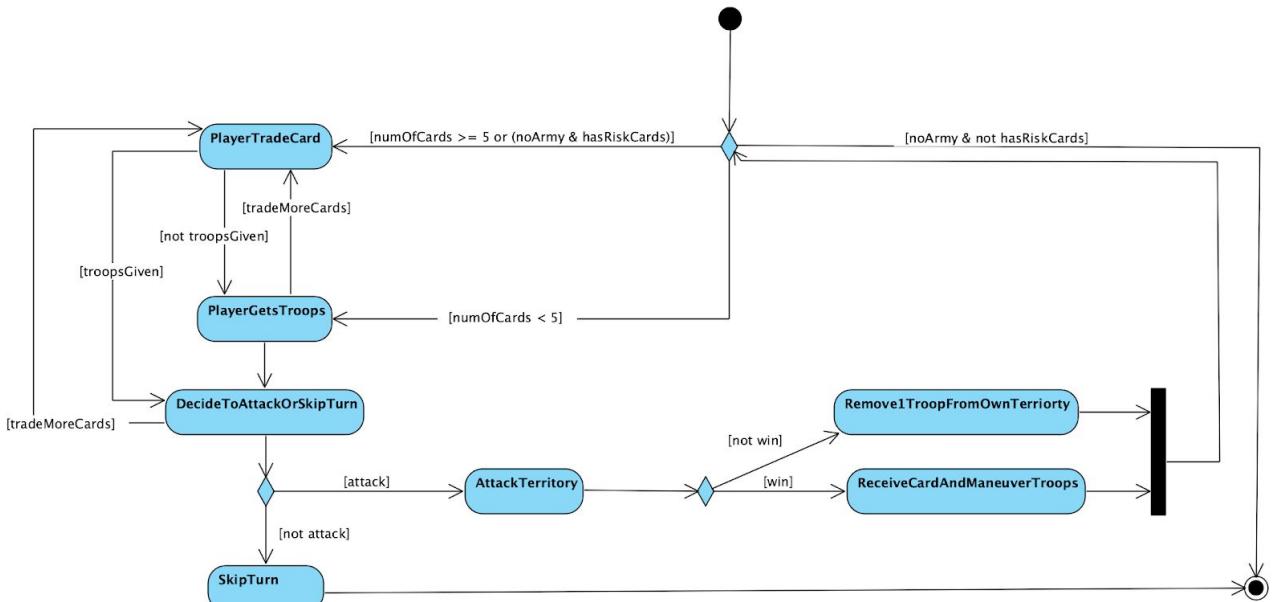


Figure 5 (Player Turn Activity Diagram)

The player turn activity diagram is the sequence of activities for the activities triggered upon a player getting a turn. This diagram indicates activities for one player and it is the same for every other player. Upon entering the activity diagram, a decision node is reached where if a player has a number of Risk cards greater than

or equal to 5 or if he has no army has a set of Risk cards that are tradable for troops, he has to trade his set of Risk cards. After PlayerTradeCard activity, we transition to the PlayerGetsTroops activity in which the player gets troops according to the rules of the game. After this activity, we transition to the DecideToAttackOrSkipTurn activity. This activity is also reached if originally the number of Risk cards is less than 5. The player can go back and trade more cards after which the player will transition back to the DecideToAttackOrSkipTurn Activity again. During this activity, the player decides if he wants to attack or not to attack. Upon reaching the decision node, if the player chooses not to attack, he transitions to the SkipTurn activity, which means that the player skips his turn and reaches the final activity. However, if the player decides to attack, he reaches the AttackTerritory activity. In this activity, the player decides on the territory to attack according to the rules of the game. A decision node is reached according to the result of the attack. If it is a win, the player receives a Risk card and has to maneuver his troops, as indicated by ReceiveCardAndManeuverTroops activity. If the player loses the attack, the player reaches the Remove1TroopFromOwnTerritory activity. In this activity, the player has to remove 1 troop from his originally occupied territory. Both activities, ReceiveCardAndManeuverTroops and Remove1TroopFromOwnTerritory, then reach the first decision node where the activity diagram started from. If the player runs out of troops and Risk cards, or the player skips his turn, he reaches the final activity.

## Trade Card Diagram

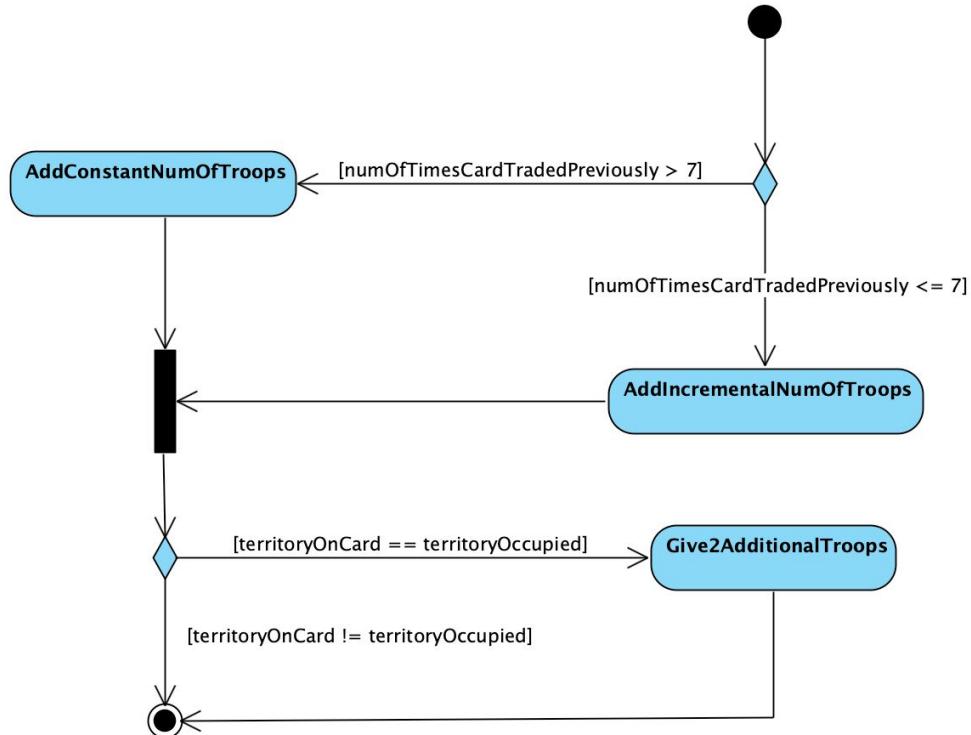


Figure 6 (Trade Card Activity Diagram)

This activity diagram represents the sequence of activities that take place when a player has to trade a valid set of Risk cards. The number of troops that can be added depends on the number of times a set of cards have been traded previously. If the number of times is greater than 7, we transition to **AddConstantNumOfTroops** activity, where 5 more troops than troops added the previous time can be added. If, on the other hand, the number of times is less than or equal to 7, the number of troops calculated can be calculated by the formula  $2n + 2$ , where  $n$  represents the number of times a set of cards have been traded previously. After these activities, if the territory on the card is the same as the territory occupied by the player, the player gets to deploy 2 additional troops. After these series of activities, the player reaches the final activity node.

### 5.2.3 State Diagrams

#### Configure Game Diagram

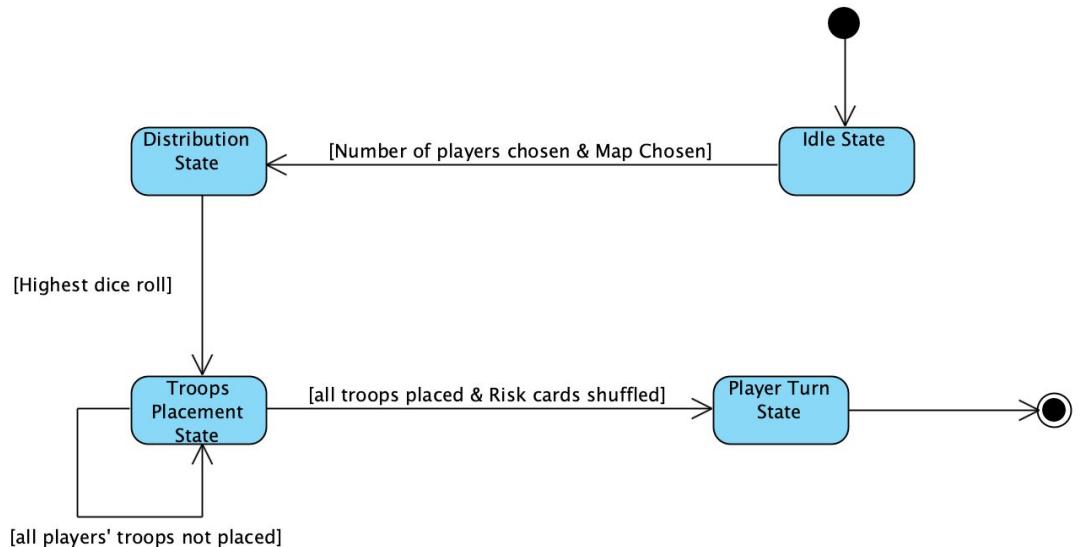


Figure 7 (Configure Game State Diagram)

Initially on game start, the game will be in an idle state until the number of players is chosen and the map is selected. The number of players can range from 3-5 inclusive. Once the number of players is chosen and the map is selected, the distribution state is reached in which the troops will be distributed to each player. The number of troops for each player will be calculated based on the formula  $50 - 5n$ , where  $n$  is a number between 3 and 5 inclusive. In the distribution state, a dice will be rolled by each player and the player who scores the highest number gets the first turn. When the highest number is scored on the dice, the state transitions to the troops placement state. In this state, each player places 1 troop in an unoccupied country one-by-one, starting from the player who scored the highest number on the dice and then continuing counterclockwise. A country can have 2 or more troops only if all other countries are already occupied. When all players' troops have been positioned in countries, a state transition occurs to Player Turn State which signifies that now the player who scored the highest die score should take his turn.

## Player Turn Diagram

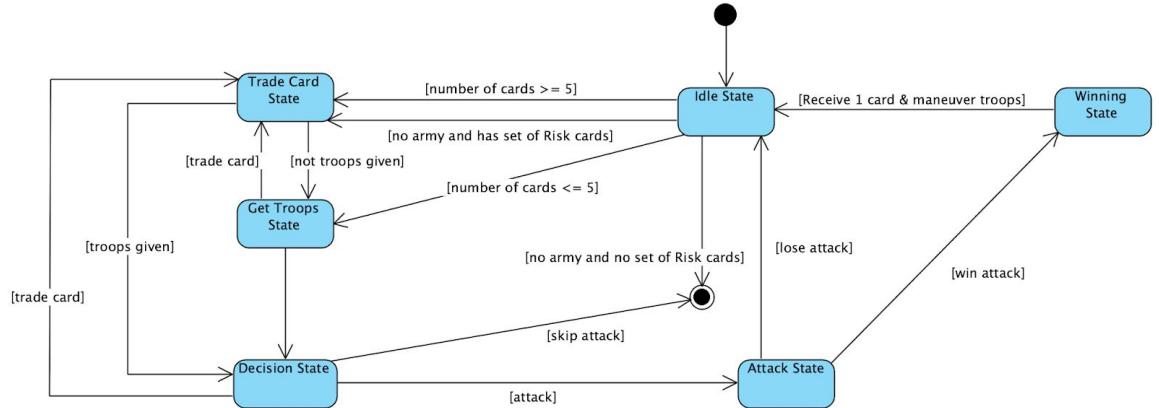


Figure 8 (Player Turn State Diagram)

On a player getting his turn, the initial state is the idle state. The current number of cards and army that the player has signifies the next state transition for the player. If the player has more than 5 cards, the state transitions to the Trade Card State. In this state, the player has to trade as many cards as possible such that at the end of the trade(s), the player has no more than 4 cards. The player will also transition from the initial state to Trade Card State if he has no army and a valid set of Risk cards as determined by the rules. If a player, however, has no army left and no set of valid Risk cards, the player is knocked out. After the Trade Card State, the player transitions to the Get Troops State, if they have previously not received troops, where the player is given troops according to the number of territories occupied by them (as per the game rules). If the player decides to trade a card in Get Troops State, he goes back to the Trade Card State and then if the player has already been given troops before for the turn, they transition directly back to the decision state. In the Decision State, the player can still go back to the Trade Card State if he wants to trade more cards. The player transitions from the Get Troops State to the Decision State. If the player originally had less than 5 cards, the player would have transitioned immediately to the Get Troops State. The Decision State is where the player has to decide if he is going to attack or skip his attack. If the player skips his attack, he reaches the final state and the next player continues his turn in a

similar way. If the player decides to attack, he reaches the Attack State. Here according to the rules dictated by the game, the player either wins or loses an attack. If the player loses, he reaches the idle state again where he is free to go through the series of events. If the player wins and reaches the Winning State, he receives 1 Risk card and has to maneuver his troops into the new territory as dictated by the rules of the game. As these two conditions are met, the player then transitions to the idle state.

### Attack Diagram

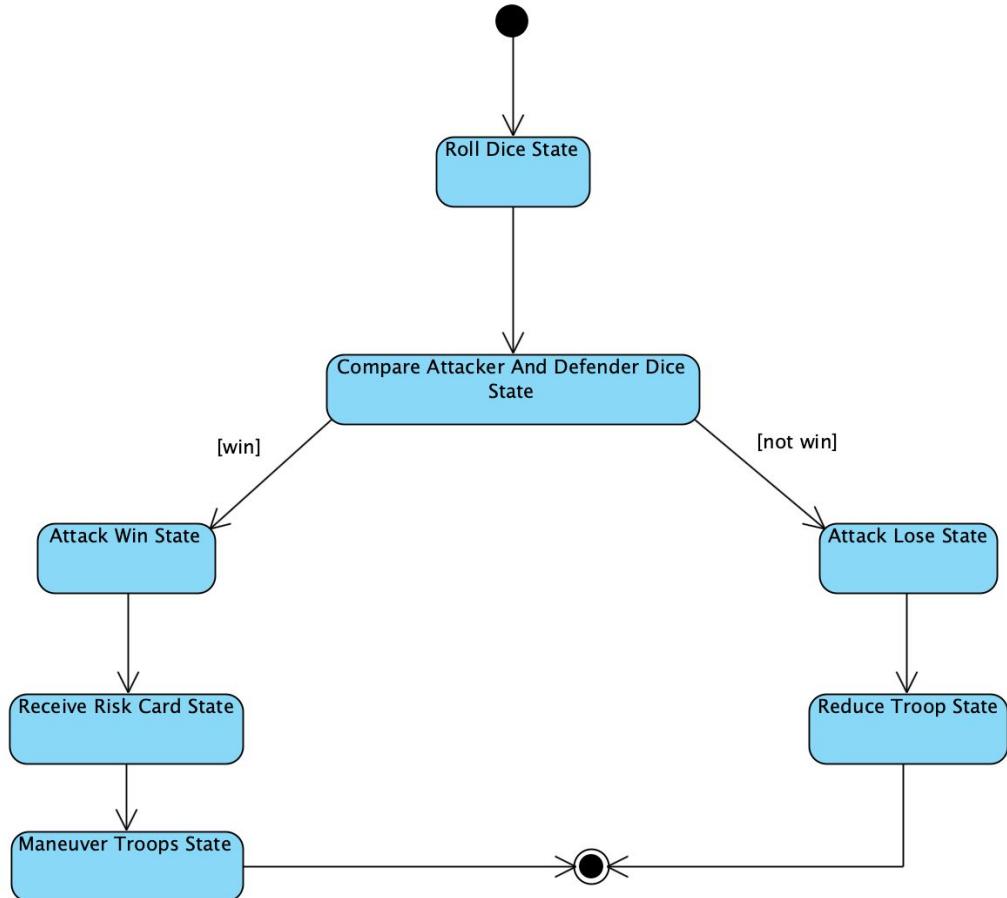


Figure 9 (Attack State Diagram)

Upon choosing to attack, the player and the defender enter the Roll Dice state. This is where the defender and the attacker roll dice according to the game rules. In the Compare Attack And Defender Dice, the dice for attacker and defender are compared. If the attack is a win, the attacker reaches the Attack Win State, else the attacker reaches the Attack Lose State. In the Attack Win State, 2 states follow, namely Receive Risk Card State – the player receives a Risk Card – and Maneuver Troops State – the player has to

adjust their troops into the newly occupied territory, leaving behind at least 1 troop in their original territory. In case the attacking player loses, the player reaches Reduce Troop State in which the player is required to reduce 1 of their troops.

### Trade Card Diagram

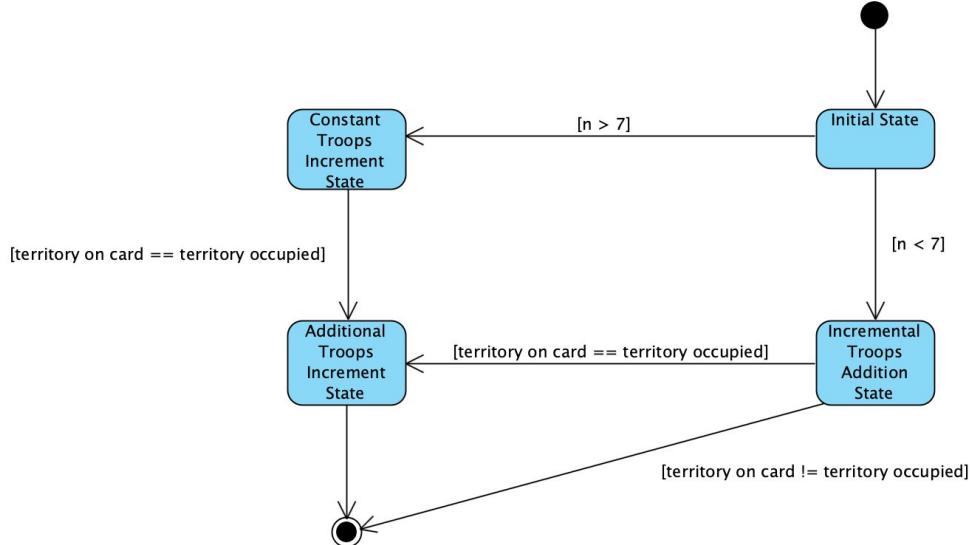


Figure 10 (Trade Card State Diagram)

When trading a Risk card, the number of times cards were traded previously affect the number of troops gained from the trade. In the state diagram above,  $n$  represents the number of times cards were traded previously. If  $n$  is less than 7 the state transitions to Incremental Troops Addition State, and the number of troops the player is allowed to add is calculated by the formula  $2n + 2$ . If  $n$  is greater than 7 the state transitions from the initial state to the Constant Troops Increment State, and a number of troops 5 greater than the previous trade can be added. If the territory on the card is exactly the same as the one the player occupies, the state transitions to Additional Troops Increment State. Here the player is allowed to add 2 troops in addition to the addable number of troops previously calculated in previous states. However, if the territories do not match, the state transitions to the final state.

## 5.3 Object and Class Model

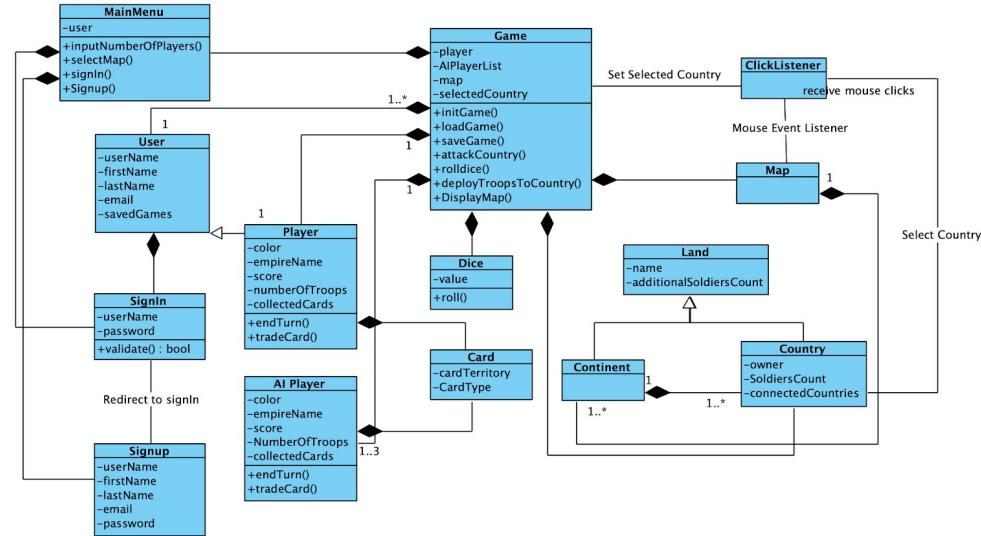


Figure 11 (Object and Class Model)

This diagram explains the class structure of the game and the components used in the game. The main class that controls everything in the game is the **Game** class. This diagram assumes that constructors, getters, setters, and destructors are present for every class.

The **Game** class is the main class controlling the program. It will get initialized when the game starts and it will let the player start a new game. It will be responsible for the gameplay rolling the dice, attacking a country, deploying troops in a country, saving the game, and loading a new game. It has the **ClickListener** attached to it to get the feedback from the user and act depending on it.

The **MainMenu** class is responsible for getting the initial user feedback on what kind of game they want to play. It also serves as the initial place for the user to signIn or signUp.

The **Map** class houses all the countries and the continents of the game. It will be attached to the **ClickListener** to check when the user has clicked on a country.

The **Land** class is the parent class for a continent and a country. Every land has a name and additionalSoldiersCount which is the

number of troops to be given to the user when this land is concurred.

The User class is the main class responsible for user management. It will validate the user's information when signing in and store them.

The Player class is a child of the User class and is game dependent. It can trade cards and end its turn.

The AI Player class is the same as the Player class but without being a child of the User class as it is an AI Player. It can trade cards, play its turn, and end it.

The Card class represents the generic Risk territory cards. It has a cardType and a CardTerritory.

The ClickListener class helps listen to any activity on the class they have been attached to and is used to interact with the user and play the game.

## 5.4 User Interface



Figure 12 (Main Menu UI Design)

Source: Adapted from [2]

This is the main menu of the game which will be initially presented to the user. The user will be required to press the login button to proceed.

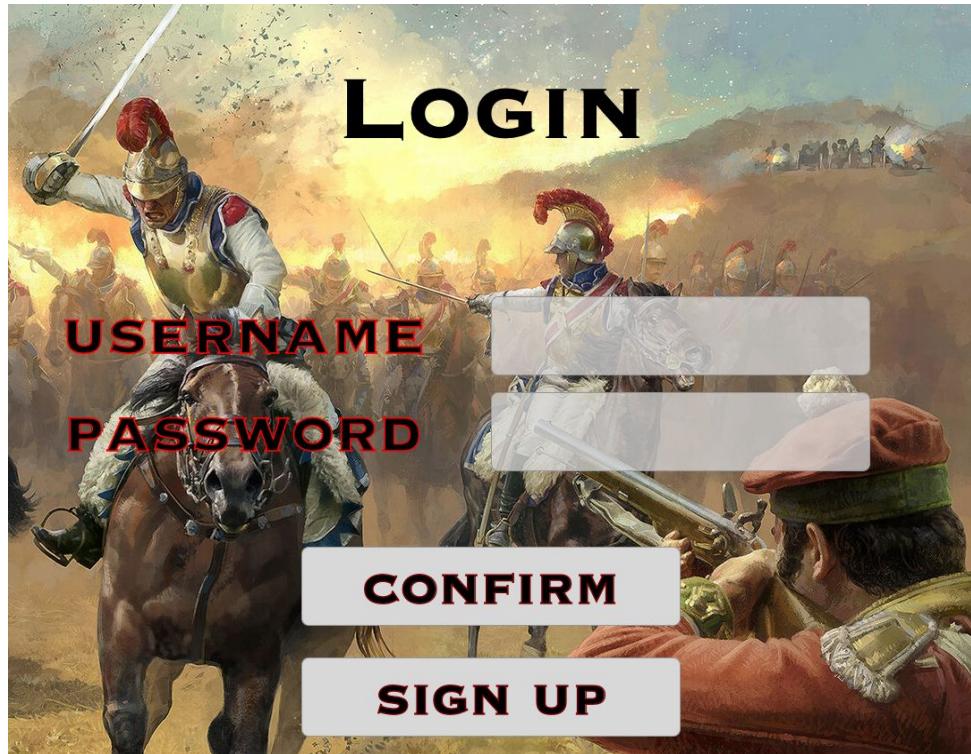


Figure 13 (Login Screen UI Design)

Source: Adapted from [2]

The login screen will prompt the user to enter his/her username and passcode in order to play, or will allow the username to sign up and create a new account.

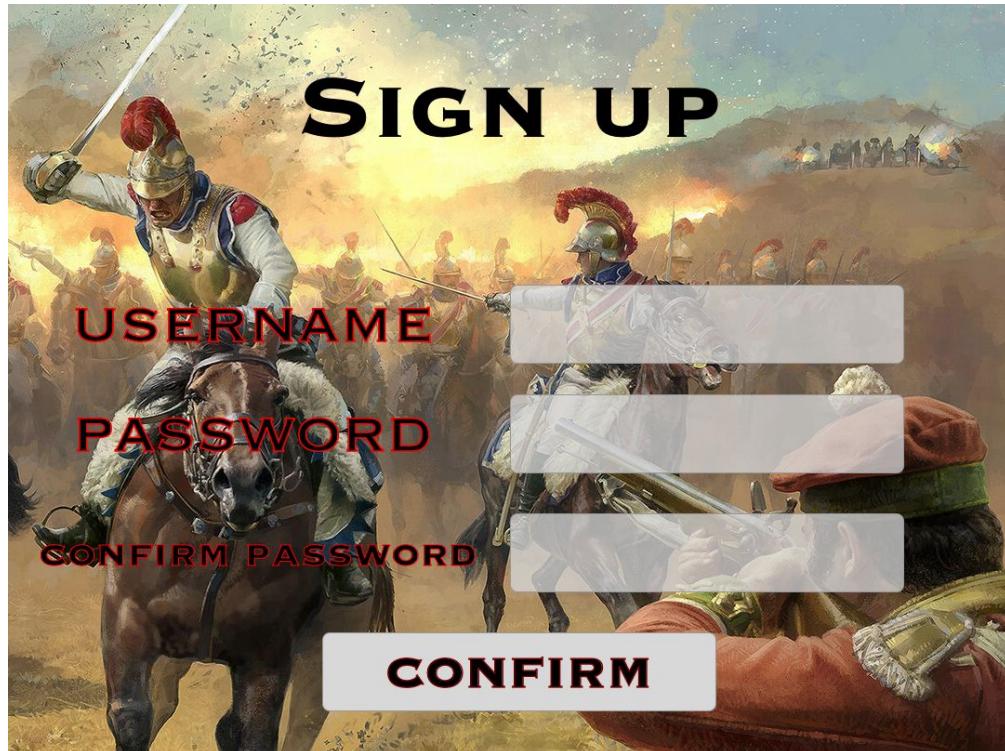


Figure 14 (Signup Screen UI Design)

Source: Adapted from [2]

The sign up screen will allow the user to create and save a new account. The user will be asked to enter his/her password twice in an attempt to avoid mistakes while typing.



Figure 15 (Launch Game Screen UI Design)

Source: Adapted from [2]

Once the user has logged in, he/she will be given the option to either start a new game or continue a previously saved game.



Figure 16 (Saved Games Screen UI Design)

Source: Adapted from [2]

Saved games screen will hold the data of previously saved and played games.



Figure 17 (Game Options Screen UI Design)  
Source: Adapted from [2]

In case the user decides to start a new game, he/she will be prompted to enter the number of players to play with. The number of players ranges from 3 to 5 inclusive.

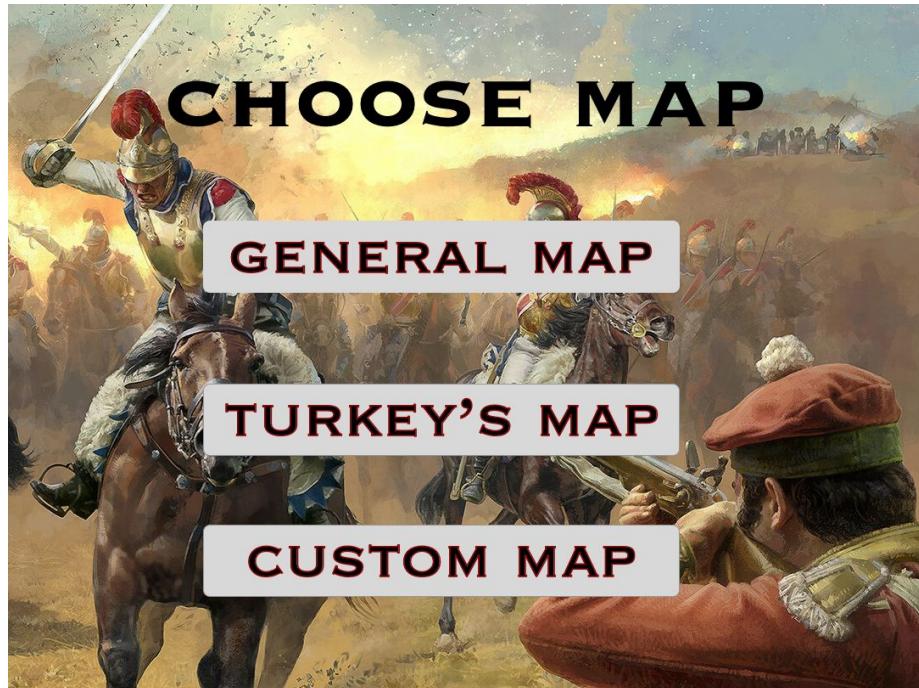


Figure 18 (Map Options Screen UI Design)

Source: Adapted from [2]

Once the number of players have been set, the user will be prompted to choose a map from a plethora of options. Some default options will consist of a general map, the map of a few countries (e.g Turkey).

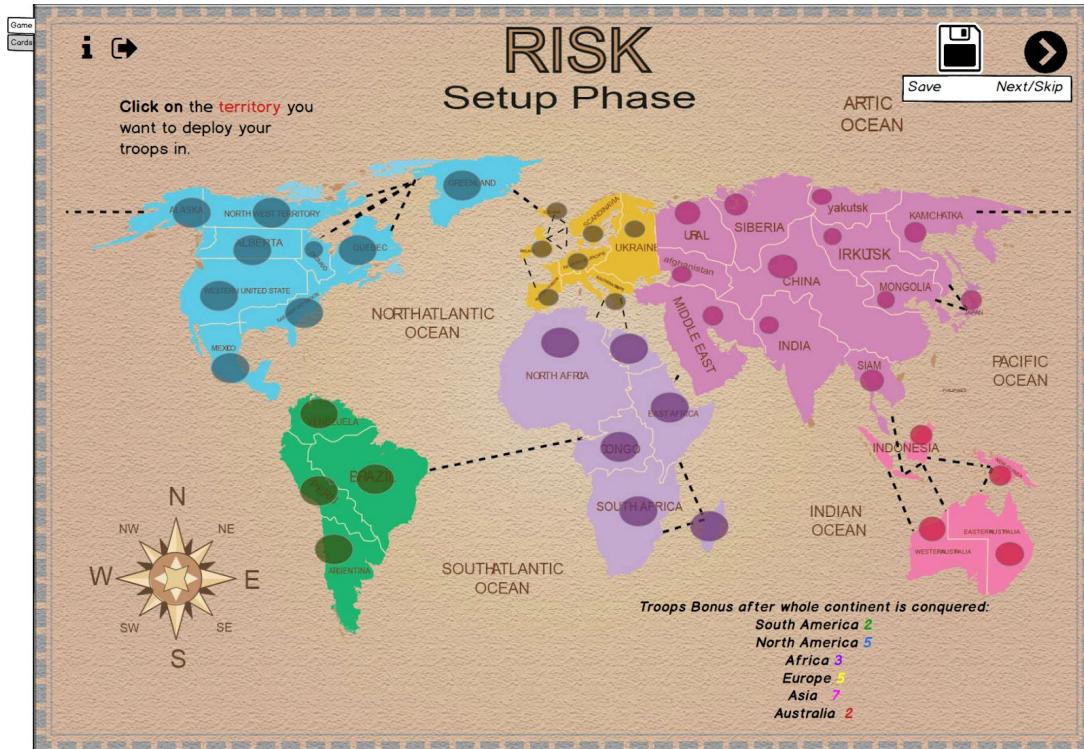


Figure 19 (Main Game Screen UI Design)

Upon choosing the general map option, the players will be required to deploy their troops. The deployable regions are indicated by dark circles of the respective color. This will be applicable for all players. The dark circles will be dynamic hence helping the players choose regions. The save and next/skip buttons on top-right of the screen would save the game and skip the player's turn respectively. The 2 tabs on the top-left, Game and Cards, are 2 screens of the game. The Game tab will present the game while the Cards tab will show the collection of cards for each player.

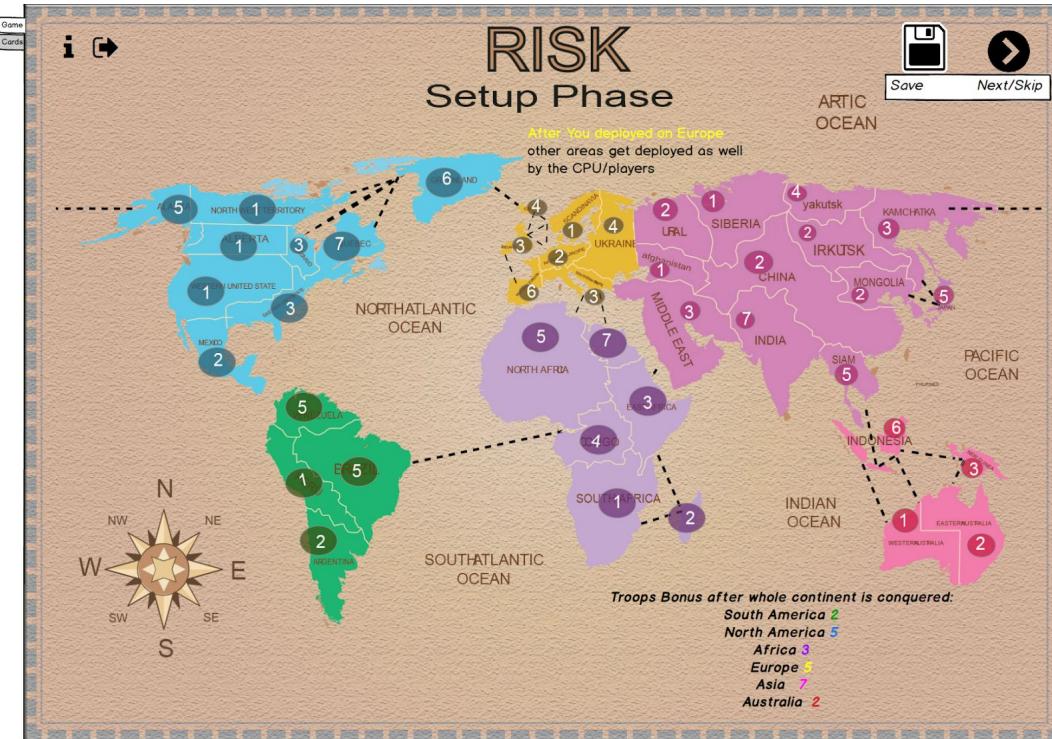


Figure 20 (Map with Troops Screen UI Design)

This is the map after troops have been deployed by each player.



Figure 21 (Map with Player Turn Screen UI Design)

When it is a player's turn, the player selects one of his/her territories. Upon this selection, the player will be presented with options for movement. The red arrows indicate a region to attack while the white ones indicate the region where a player can maneuver his/her troops.

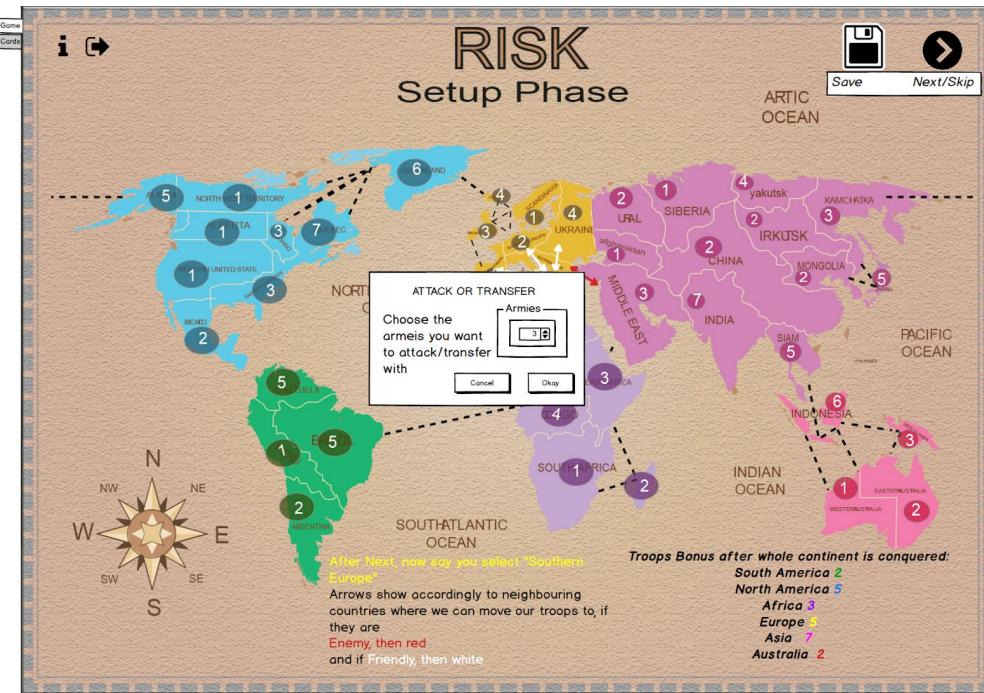


Figure 22 (Attach Screen UI Design)

When the player chooses an option, he/she will be presented with a popup to choose the number of troops he wants to use while attacking or moving.



Figure 23 (Dice Screen UI Design)

After choosing, if the player decides to attack, 2 sets of dice will be rolled according to the situation of the attack. Red dice are rolled for the attacker and the white for the defender.



Figure 24 (Card awarding Screen UI Design)

If the dice roll results in the win of the attacker, the attacker is awarded a card.

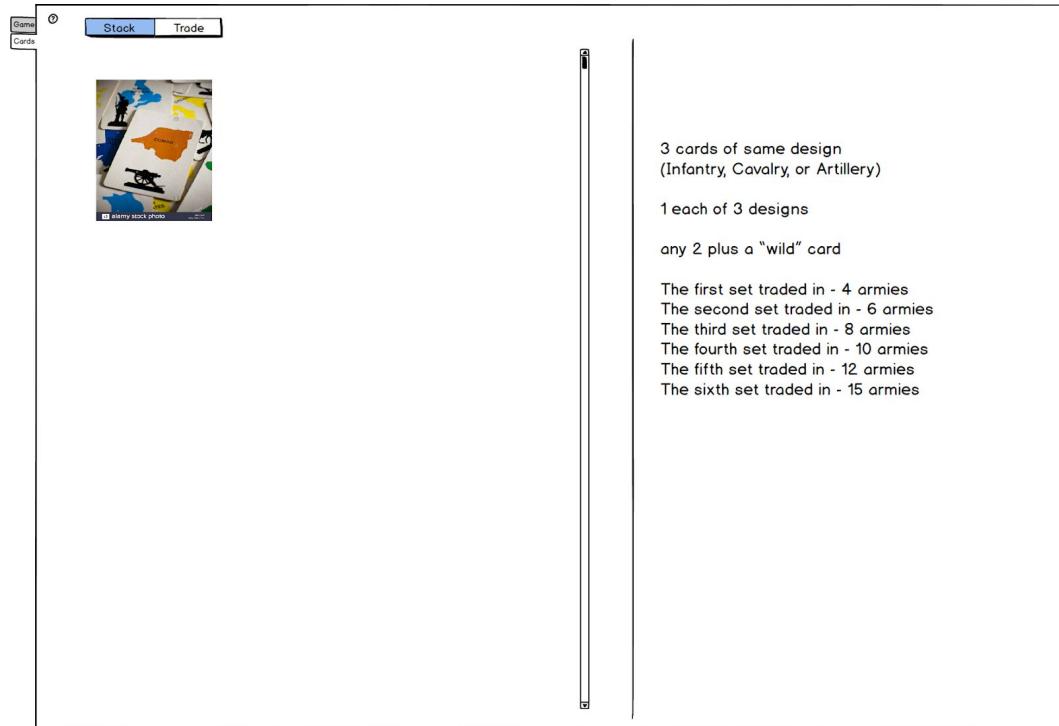


Figure 25 (Card Description Screen UI Design)

The awarded card is shown up in the Cards tab on the top-left of the screen. The player is also presented with a set of instructions of the cards that can be traded.



Figure 26 (Conquered Continent Screen UI Design)

In this case, a continent is conquered.

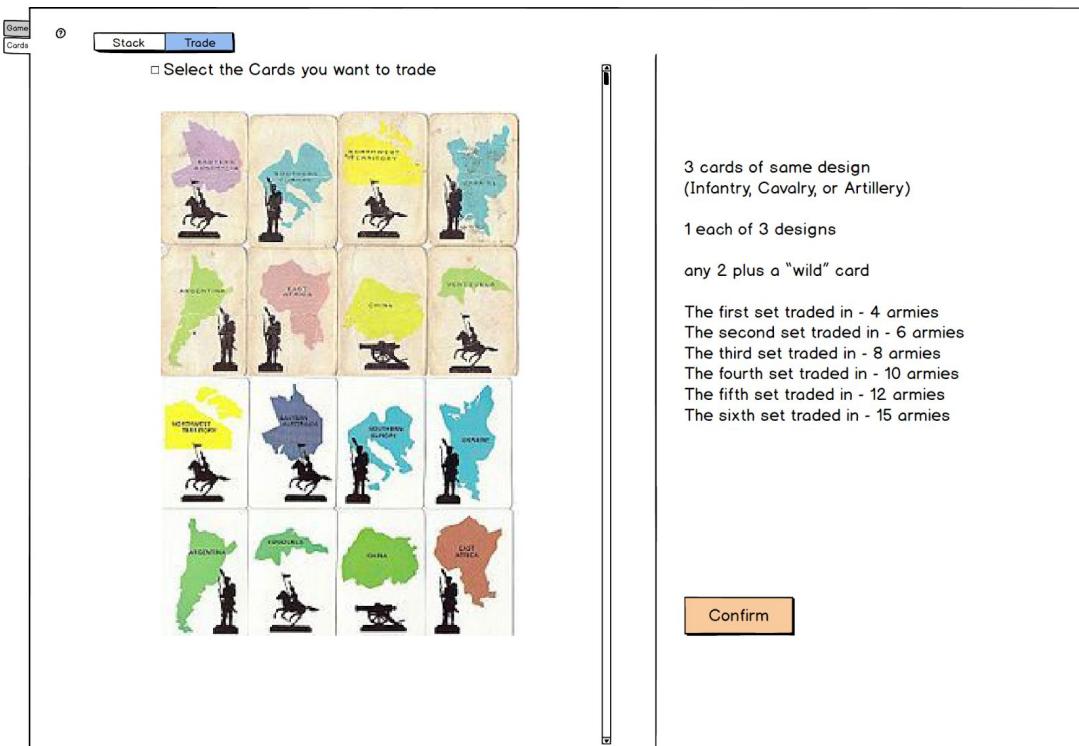


Figure 27 (Cards Description Screen UI Design)

This is the screen where the player will be able to trade a set of cards.



Figure 28 (Save Game Screen UI Design)

Source: Adapted from [2]

In case the top-right pause button is pressed during the gameplay, this screen will show up and will prompt the user to save the game. In case the user chooses yes, the game's data will be recorded in previously saved games.

## 6. Glossary and References

[1]"Risk (game)", *En.wikipedia.org*, 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Risk\\_\(game\)](https://en.wikipedia.org/wiki/Risk_(game)). [Accessed: 30- Oct- 2020].

[2]S. Beliaev, *Cavalry Horse*. 2020.