SOEN 6441: Advanced Programming Practice

Winter 2019

Build 3

Submitted To: **Amin Ranj Bar**

Submitted By: **Team 18**

Sr. No.	Name	Student ID
1	Nikitha Jayant Bangera	40088393
2	Shresthi Garg	40105918
3	Anusha Keralapura Thandavamurthy	40102962
4	Shiva Shamloo	40094742

Architecture for Risk Game – Model-View-Controller

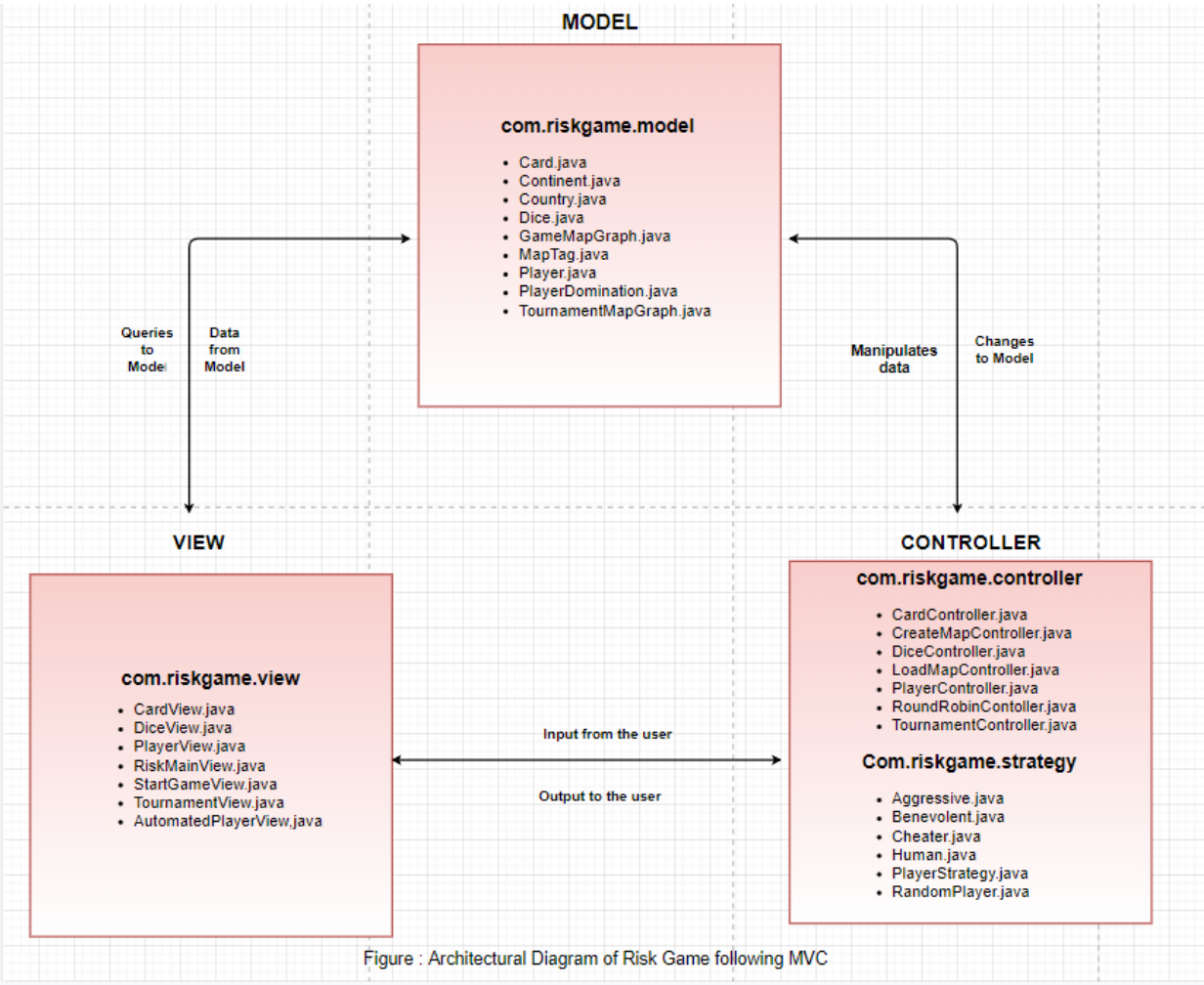

Figure : Architectural Diagram of Risk Game following MVC

Following are description for each layer

1. Model – Has core classes for the game and related logic

2. View – Has classes related to view

3. Controller – Acts as a bridge between model and view.

Modules in View: We have seven view classes with following functionality.

1. **CardView Class**: This Class aims to show the card view. It will display the type of cards available with the player and it also contains exchange button which is used to exchange the cards with armies.
2. **DiceView Class:** This Class is used for dice view where player can see his dice count. It displays the attacker and defender information along with the number of dices. It also has roll dice button which is used during attack.
3. **RiskMainView Class:** Class used to provide an interface for the players to interact with the game. It launches the game and window is displayed where user can create or load map to begin the game.

4. **PlayerView Class:** This Class gives the player view. It will display all the information related to the current player. It shows the countries belonging to the player along with its adjacencies. It also gives player domination view. PlayerView has buttons such as place army, reinforcement, attack, all out, fortify and end turn.
5. **StartGameView Class:** This Class gives the start view of the game, it lets the players choose the number of players that wishes to play the game, their name and the strategy they wish to play the game with viz., Human, Benevolent, Cheater, Aggressive
6. **TournamentView Class:** This Class gives the tournament view and lets the player select number of maps and player and their specific details, number of games and turns between 10-50 as the range.
7. **AutomatedPlayerView Class:** This view pops up when none of the player who have chosen to play the game has the player type as Human.

Modules in Controller: We have thirteen classes in controller with the following functionality**.**

1. **CardController:** This class is used to allocate the cards to the players. It has the methods which assigns cards to the country and allocates a card randomly to the player if the player wins the country.
2. **CreateMapController**: This class is used to create a connected from the user input and provides a method to modify the map data.
3. **DiceController:** This class aims to assign dice to the player based on the armies and then if the attacker wins it lets him move the armies to the defender's countries.
4. **PlayerController:** Player Controller is for the beginning of the game play. It contains methods which will take the details from the players and starts the eventual phase. It has methods such as allocationOfArmyToPlayers(), reinforcementPhase(), attackPhase() etc.
5. **LoadMapController:** This class is used to write all the map details to the file and read the contents of the file to validate or edit.
6. **RoundRobinController:** This class provides functionality for round robin traversal among the players of the game.
7. **TournamentController:** This class controls the actions related to the tournament. It is responsible for initiating and completing valid tournament based on the player's input.
8. **Aggressive**: It bears the strategy that governs aggressive players game phases
9. **Benevolent**: It bears the strategy that governs benevolent players game phases
10. **Cheater**: It bears the strategy that governs cheater players game phases
11. **Human**: It bears the strategy that governs human players game phases
12. **PlayerStrategy**: It bears the strategy that governs aggressive players game phases
13. **RandomPlayer**: It bears the strategy that governs random players game phases

Modules in Model: We have eight model classes which have following functionality.

1. **Card:** This class is a model for the card that is owned by the players in the game
2. **Continent:** This class model the continent. It is a set of countries where each country belongs

to only one continent.

3. **Country:** Used to model the country. Each player will be assigned with few countries and player aims at occupying all the countries.
4. **Dice**: This class aims to model the dice of the players in the game.
5. **GameMapGraph:** Class used to model the map. It has all the details of the map and its content.
6. **MapTag:** Used to model the Map tag data present in map.
7. **Player:** Used to model the players of the game. It has all the details related to the players.
8. **PlayerDomination:** This class is a model and aims to find the domination view of the player in the game. It has method which computes the domination view and returns the same to be displayed for each player.
9. **TournamentMapGraph:** Class used to model the details of players, number of players , number of maps, games and their respective result in a tournament. It has all the details of the tournament and its content.