DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE

ENGINEERING

SOEN 6441, Fall 2019

RISK Game (Build-2)

Architecture Design

Submitted To: JABABO KHALED

Submitted By: **Team E**

Git URL: https://github.com/Surya64/APP_SOEN-6441_TeamE

Sr. No.	Name	Student ID
1	Surya Prakash Govindaraju	40085527
2	Shruthi Kondapura Venkataiah	40091427
3	Sahana Anantha	40085533
4	Sai Charan Teja Doddi	40076338
5	Dolly Modha	40084358

1. Introduction:

We have developed the project on RISK game by following the Model View Controller (MVC) software design architecture style

2. Architecture Design

Model View Controller Architecture (MVC):

Architecture Diagram of RISK Game development: MVC architectural style

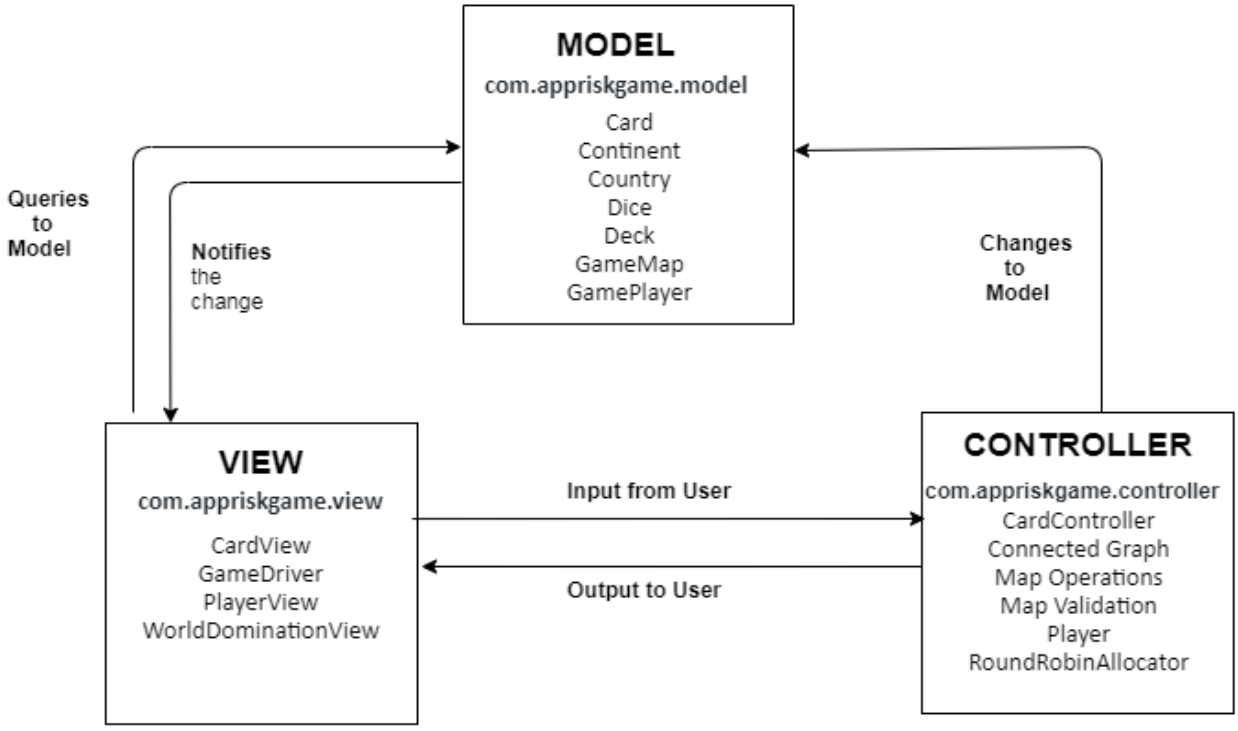

Fig : MVC Architecture Diagram of RISK development Project

Following the MVC design pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts. This is done to separate the internal representation of information from the ways the information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

Each Layer is described as below:

1. **Model** : Contains the classes which holds the logic of the Risk game
2. **View**: Contains the class which decided where the controls and display go.
3. **Controller** : Contains the classes which bridges between Model and View

3. Modules in each Layer.

Modules in View: We have four 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. **GameDriver**: Class used to provide an interface for the players to interact with the game. It launches the game and in console the user can create or load map to begin the game.

3. **PlayerView**: 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.

4. **WorldDominationView**: This Class is used for implementation of World Domination View. It gets notified whenever there is an update to the game player

Modules in Controller: We have the below classes.

1. **ConnectedGraph**: This class is used to check the connectivity if the countries form connected graph.

2. **MapOperations**: This class contains the methods for Add/Update/Delete Continent, Country, create new map, load and edit map operations.

3. **MapValidation**: This class is used to check the validation of map which is newly created or existing map is loaded.

4. **Player**: 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 and fortification phase.

5. **RoundRobinAllocator**: This class provides functionality for round robin traversal among the players of the game.

6. **CardController**: This class contains all the methods for card allocation and exchange of cards.

Modules in Model: We have the below classes.

1. **Continent**: It contains all the information of the continent and a list of all the countries that belong to a continent

2. **Country**: This class is used to model the country. Each player will be assigned with few countries and player aims at occupying all the countries.

3. **GameMap**: Class used to model the map. It contains all the details of the map and its content.

4. **GamePlayer**: It contains all information related to a player and the number of armies assigned to the player

5. **Card**: This class is a model for the card that is owned by the players in the game

6. **Dice**: Dice class is a model which gives dice information of player in the game

7. **Deck**: Deck class is model which contains all the details of cards.