**SOEN 6441: Advanced Programming Practices**

**Winter 2019**
**Project – Risk Game**
**(Build 3)**
**Architectural Design Document**

**Submitted By:**                                    **Submitted To:**

**Team 30**                                           Amin Ranj Bar

| NAME | ID |
|------|-----|
| 1. Gargi Sharma | 40042837 |
| 2. Jaiganesh Varadharaju | 40081862 |
| 3. Md Hasibul Huq | 40087646 |
| 4.Narendran Krishnakumar | 40089619 |
| 5. Zakiya Jafrin | 40021416 |

# Table of contents

# 1. Introduction

This introduction provides an overview of the **Software Architecture Document** for the Risk Computer game. It includes the purpose, scope and overview of the whole system.

# 2. Scope

The scope for the Risk game in Build 1 according to the guidelines are:

## 2.1   Map editor

This part covers all the functionalities of Map which includes the following activities:
   a.  Create a new map
   b.  Editing an existing map
   c.  Add/delete continent
   d.  Add/delete country
   e.  Saving a map to a text file exactly as the conquest.html format
   f.  Verification of map correctness

## 2.2 Game Play

This part covers all the functionalities of Game phase which includes the following activities:
   a.  Select the valid map for the game.
   b.  Define the number of players and assign the countries in a random way.
   c.  Implementation of Phase view using Observer pattern.
   d.  Implementation of Players World Domination View by calculating the percentages, continents owned, the total number of armies owned by every player.
   e.  Implementation of the different player behaviors using the Strategy pattern, where the strategies implement different versions of the reinforcement, attack and fortification as methods of the Player class
   f.  Single game mode: Game starts by user selection of a user-saved map file, then loads the map as a connected graph. User chooses the number and behavior of players .The game proceeds until one of the players has conquered the whole map. If no human player is selected, the game proceeds fully automatically without any user interaction.
   g.  Tournament Mode: When the game starts, provide an option for a Tournament Mode (see "Tournament" below). The tournament should proceed without any user interaction and show the results of the tournament at the end.
   h.  Players are allocated a number of initial armies, depending on the number of players in a round robin fashion.

     i.   Reinforcement phase with the calculation of a correct number of reinforcement armies.

     j.   Implementing Attack Phase by deciding attacker and attacked country, a proper number of armies are deducted from attacker/defender country during attacks. 2 An attacker can move any number of armies in the conquered country if the defender is conquered.

     k.   Implementation of an "all-out" mode.

     l.   Fortification phase with an implementation of a valid fortification move.

     m.   Game Save/Load: As a game is being played, allow the user to save the game in progress to a file, and allow the user to load the game in exactly the same state as saved.

## 3.  Software Architecture

In our Project, we are using the Model-View-Controller Architecture to differentiate the classes. MVC divides the application into three different components - Model, View and Controller.

- **Model:** It is responsible for maintaining the data of the application. Its objects retrieve and store the state in a database.
- **View:** It is responsible for displaying all the data to the user. So, it is a user interface of the application.
- **Controller:** It is responsible to handle user's requests and renders the view with the model data in the form of response.
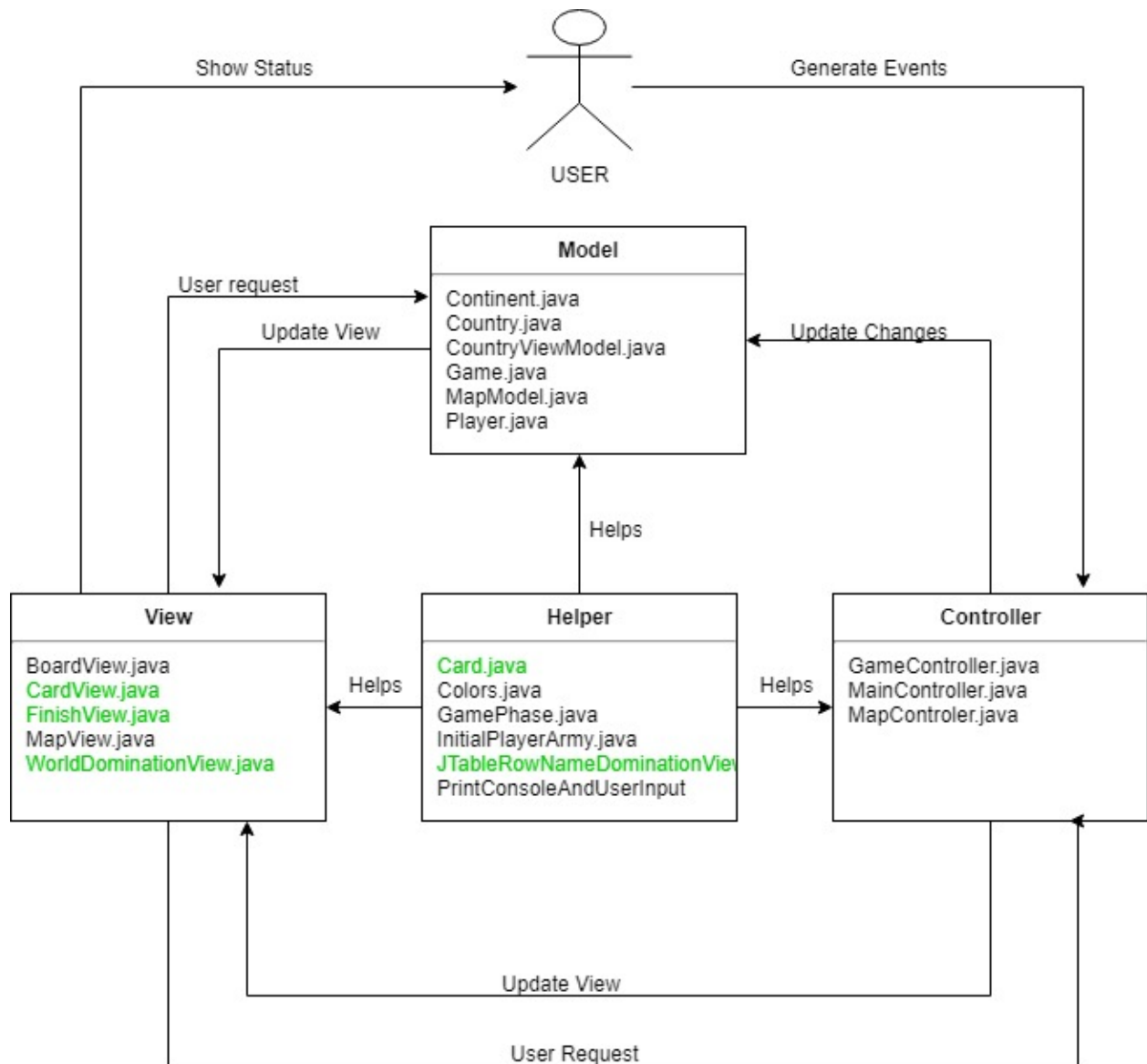
Figure 1: Project Architecture

* Newly added files for Build2 are written in green colour.

* Strategies are added for the Build3 requirements.

## 4. Modules Description

### 4.1 Controller

The controller folder includes the Main controller, Map Controller and the Game controller for the Risk Game. The description is as follows:

| Controller file name | Description |
|---|---|
| MainController | This is the main controller which runs the game and all other parts of the Risk Game. |
| MapController | This class performs the functionalities of Map which includes:<br>● Add/delete Continent<br>● Add/delete Country<br>● It updates the MapView based on the data changed by the MapModel.<br>● It serves to all the request issue by the MapView. |
| GameController | This class acts as a mediator between the Game model and the  BoardView file. It performs the following actions:<br><br>● Creation of player<br>● Assigning armies<br>● It serves to all the request issued by the BoardView. |

## 4.2 Model

The model folder includes the Map Model, continent, country, player and Game model which are connected. The description is as follows:

| Model file name | Description |
|---|---|
| MapModel | This class stores data of map file and checking the validation of the existing and editing map. |
| Player | This class has players attributes and basic setter getter functions to get and set the value out of it. Moved the reinforcement, attack and fortification methods from Game model to player class as per the guidelines of Build2. |

| Game | Game model is used to create a model for the game. It is bounded with the Game Controller and the Board View. |
| --- | --- |
| Continent | This class is used to read and store different elements of the Continent which are updated by MapController. |
| Country | This class is used to read and store different elements of the country. |
| CountryViewModel | It collects the country and player information for viewing purposes. |

### Strategies

The Strategies folder has different types of strategies which has to select by the player.

| Strategy name | Description |
| --- | --- |
| Human | A human player that requires user interaction to make decisions. |
| Aggressive | An aggressive computer player strategy that focuses on attack (reinforces its strongest country, then always attack with it until it cannot attack anymore, then fortifies in order to maximize aggregation of forces in one country). |
| Benevolent | A benevolent computer player strategy that focuses on protecting its weak countries (reinforces its weakest countries, never attacks, then fortifies in order to move armies to weaker countries). |
| Cheater | A cheater computer player strategy whose reinforce() method doubles the number of armies on all its countries, whose attack() method automatically conquers all the neighbors of all its countries, and whose fortify() method doubles the number of armies |

| | on its countries that have neighbors that belong to other players. |
|---|---|
| Random | A random computer player strategy that reinforces random a random country, attacks a random number of times a random country, and fortifies a random country, all following the standard rules for each phase. |

## 4.3 View

The view folder includes the following classes:

| View file name | Description |
|---|---|
| MapView | This class shows the map menu and view window while creating a  map from scratch. It is connected with Map controller and Map Model. |
| BoardView | This class shows the graphical user interface of the Risk Game. This board view shows the view of all game phases |
| CardView | This class is used to chose the cards by the player during the Reinforcement phase of the game to obtain new armies. |
| WorldDominationView | This class shows the updated view of the Players world domination which includes<br>(1)Percentage of the map controlled by every player.<br>(2)The continents controlled by every player.<br>(3)Total number of armies owned by every player. |
| FinishView | This class is used to display the Congratulations message to the player who wins the game. |
| TournamentModeResultView | This class is used to display the tournament result. |

**4.4 Helper**

The helper folder has common files which are used by all the other folders of the project.

| Helper file name | Description |
|---|---|
| Colors | This class is used to add colors in the game. |
| GameMode | This class is used to select Game mode when starting to play the game. |
| GamePhase | This class contains the phases of the game. |
| InitialPlayerArmy | This class contains the values for how many armies are going to get by a specific number of players. |
| PrintConsoleAndUserInput | This class has some input-output functions for various data types. |
| JTableRowNameDominationView | This class is used to create the static rows with the element name in the world domination view. |
| RandomNumber | This class generates a random number to be used for the Random Strategy |

# 5. Tools and Technologies

| Tools and Technologies | Description |
|---|---|
| Eclipse | We have used Eclipse IDE for game development. |
| Junit4 | We have used Junit4 for writing the test cases. |
| JAutodoc | We have used JAutodoc Eclipse Plugin to add Javadoc and file headers in the source code. |

## 6. References

http://www.cs.toronto.edu/~wl/teach/407/2002/rup-sad.html
Class Lecture