SOEN 6441: Advanced Programming Practices

Winter 2019
Project – Risk Game
(Build 2)
Architectural Design Document

Submitted By:
Team 30

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

Submitted To:
Amin Ranj Bar

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. Players are allocated a number of initial armies, depending on the number of players in a round robin fashion.
f. Reinforcement phase with the calculation of a correct number of reinforcement armies.
g. 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.
h. Implementation of an “all-out” mode.
i. Fortification phase with an implementation of a valid fortification move.

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.

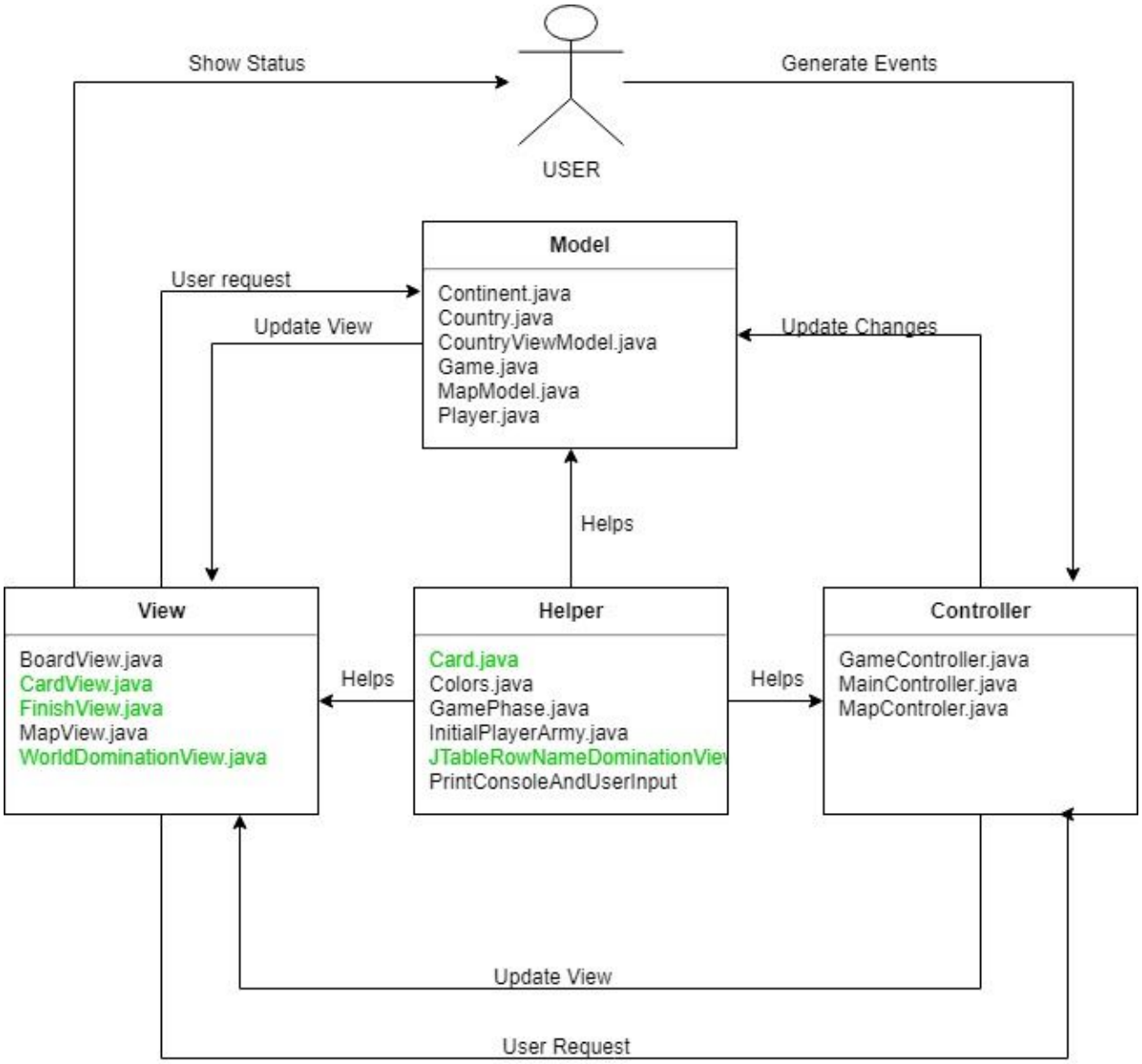

Figure 1: Project Architecture

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

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: ● Add/delete Continent ● Add/delete Country ● It updates the MapView based on the data changed by the MapModel. ● 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: ● Creation of player ● Assigning armies ● 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.

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 (1)Percentage of the map controlled by every player. (2)The continents controlled by every player. (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.

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.
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.

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