Department of Engineering and Computer Science

Concordia University

Montreal, Quebec, Canada

Advanced Programming Practices
(SOEN-6441)

(Winter 2019)

Architectural Design
Project: Risk: Strategy Build-1

Team-35

Name	***Student IDS***
Gursharan Deep	*40039988*
Harmanpreet Singh	*40059358*
Raghav Sharda	*40053703*
Yaswant Choudary Narra	*40087595*
Saman Soltani	*40093581*

Architecture Style Used in Project:

For our project, we have implemented **Model-View-Controller Architecture** (MVC) for the implementation of user Interface.

Brief Description of MVC Model:

MVC - is three-tier architectural model and it is widely used for many object-oriented designs with user interaction. MVC separates application into three components - Model, View and Controller.

Model: Model represents handling of the data and business logic. It maintains the main data of the system.

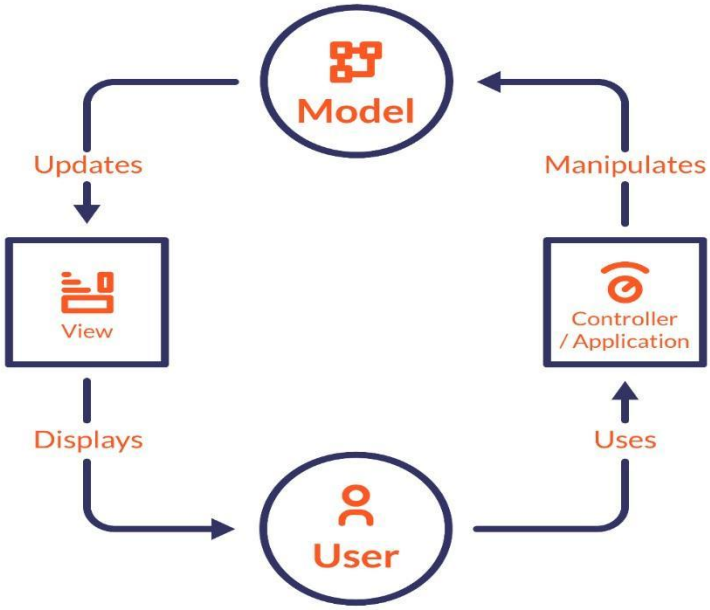
Model is a data and business logic.

View: View is a platform for user interaction. View display data using **Model** to the user and enables hi, to update the state of the data.

View is a User Interface.

Controller: Controller handles the requests from user through interaction with **View** and make **Model** change its data state. The controller renders the appropriate view with the model data as a response.

Controller is a request handler from **View**



MVC Model Architecture:

SCOPE :

The scope of this document covers following **functional requirements** as per the instruction set provided for the build:

- **Map Editor**: It allows user-driven creation of new map and modifying an existing map split into following two parts:
    - **NewMapCreation**: It deals with creation of new map by user by defining map elements like-country, continent and connectivity between them.
    - **ExistingMapModifier:** It deals with modifying an existing “conquest” map file and add/delete/update continent/country/neighbor country.
- **Game Play:** It deals with implementation of game phases as per the defined risk game rules and split into following parts:
    - Selection of saved maps (connected graph), selection of number of player and random assignment of countries (**startup phase**), allocation of initial armies and placing armies one by one.
    - Calculation and placement of reinforcement armies as per rules. (**Reinforcement Phase**)
    - Fortification moves are made as per risk rules (Fortification Phase).

Architecture Description-Project

The whole application of risk game (build-1) is divided into three interrelated layers hiding internal representation from user interaction with the system. It allows **code reuse.** Following is description of main modules in risk Game:

MODEL:

This part manages behavior of the game on interaction of user with the system. Our project implements different models handling **game data** following observer pattern (observable) and modify data of game as per requests made by user. Different parts handle state of different game entities like card information, game phases, maps, player information etc, explained below:

ArmyCount: This class is initializing the number of armies based on number of players.

Card: This class represents the Card Model and contains information about assigning and trading card betwP Nrenaeen players

Dice: This class represents the Dice Model and perform dice operations like -Assignment of values to each dice randomly. Validation before dice rolls. Assignment of Territory and move armies when player won.

GameDriver: It provides data like player information, maps and player setup during game play

GamePhase: This class is used for Attack, Reinforcement, Fortification.

Map: The Map Model deals with all data related to conquest maps used in game implemented as **observable** for views. Handles map data and map Editor.

NodeOfCountry: store all relevant data relating country.

NodeOfMap: Map data is updated/stored using this class.

Player: This class is responsible to represent the player

ReadMap: used to read map file from user defined location.

WriteMap: used to write a map file to user defined location.

VIEW:

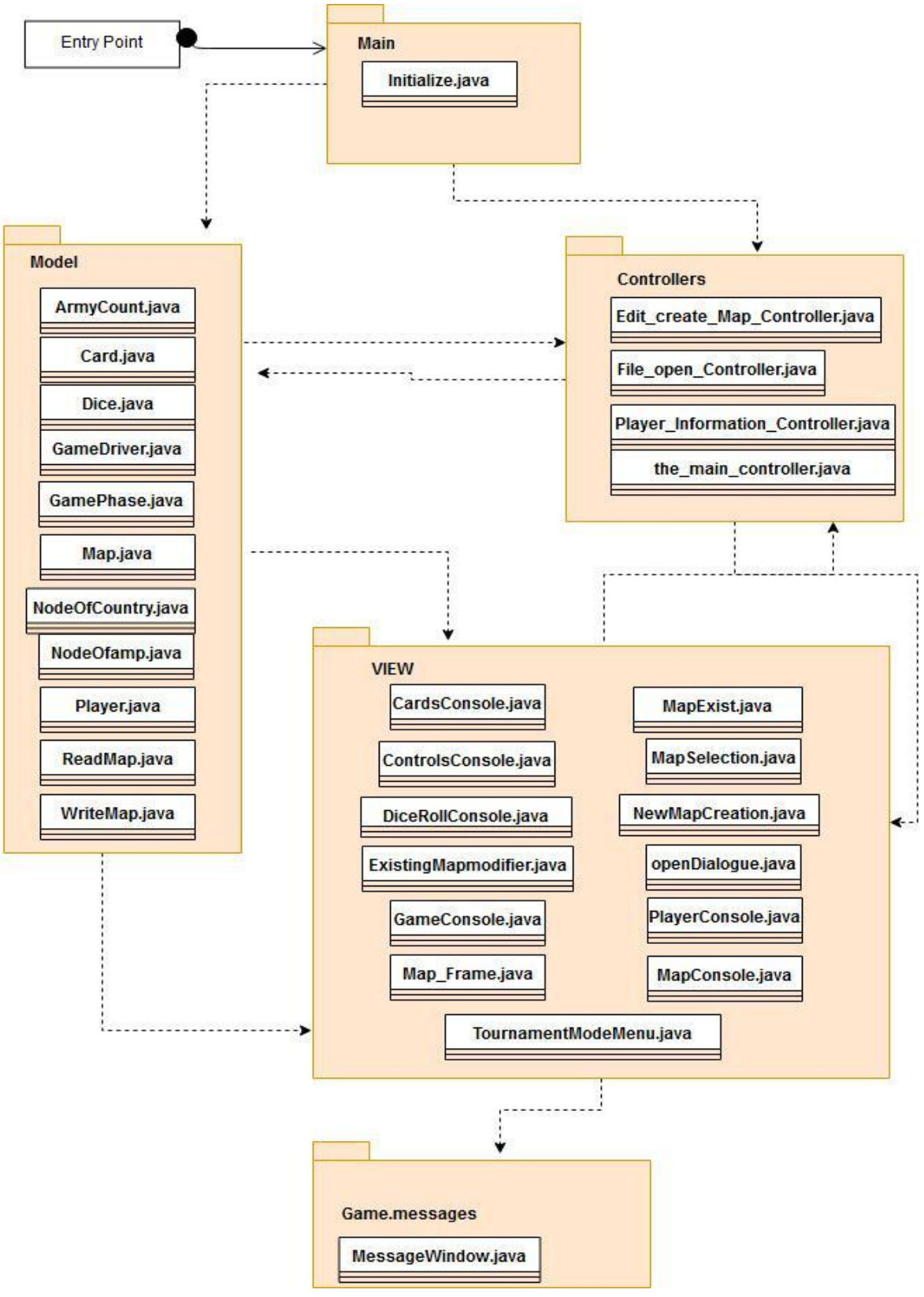
It deals with different user interface platform allowing user interaction. There are multiple views in the project and all of them are implemented as **an Observer pattern:**

CardsConsole: this class creates the cards view on main game console

ControlsConsole: this class creates the view of various game controls like army count etc. on main game console

ExistingMapModifier: class opens the JFrame view for delete/add country and continent while modifying Map

GameConsole: This class is the **main view** of the front end of the game console where the players will play the game and all other information while playing the game will be displayed



Package Diagram- (Representing Architecture Elements of Risk Game)

Map_Frame: opens JFrame view for selecting either a New map or an Existing map.

MapConsole: set coordinates for map.

MapExist: This class opens JFrame to open map file and initialize contents of frame

MapSelection: It implements view for map file selection

NewMapCreation: NewMapCreation class opens the JFrame view for creating new map, add/delete country and continent

Openingdialog: It is main View of the game that opens up when application is initialized.

PlayerConsole: this class creates the view for Player information display.

Controllers

These classes issue directions to Model classes to update themselves as per user interaction. It handles main game-logic and coordinate communication between VIEW and MODEL. It contains following modules:

the_main_controller: controller class which controls the main functioning of the game in the initial phase.

Edit_create_map_controler: controller class that perform actions for creation of new map and modification of existing map. Responds to requests of NewMapCreation and ExistingMapModifier.

File_open_controller: control the opening of various files from local system during game.

Player_information_controller: This class will return all the information required for a player-the number of players and name of players