RISK GAME

Architecture Design Document

Advanced Programming Practices (SOEN 6441)

Winter 2019

Team No. 41

Sidhant Gupta	40059256
Shivam Patel	40093311
Sharvesh Vora	40081458
Jil Mehta	40089806
Nagmeh Ansari	40108475

Architecture Design Document

1. Overview

   RISK game is turn based strategy game where each player tries to conquer all countries on the map.

   Game can be divided into 3 phases

   a. Reinforcement – Player is able to add new armies to his countries
   b. Attack – Player can choose to attack countries owned by another player
   c. Fortification – Player can move armies between their countries to fortify them

2. Technologies Used
   a. Java Swing – handling and rendering the UI elements
   b. Junit 4 - implementation of unit test cases
   c. JGraphT – Library which provides various functionalities for handling Graphs

Requirements:

Map Editor

1. User-driven creation of map elements, such as country, continent, and connectivity between countries.
   a. User should be able to add Continents
   b. User should be able to add Countries
   c. User should be able to assign a country to a continent
   d. User should be able to add neighbors of a country

2. Saving a map to a text file exactly as edited (using the "conquest" game map format)
   a. User should be able to save a valid map as a map file as edited in the map editor

3. Loading a map from an existing "conquest" map file, then editing the map, or create a new map from scratch
   a. User should be able to load an existing map file
   b. User should be able to edit an existing map file
   c. User should be able to create a new map

4. Verification of map correctness upon loading and before saving (at least 3 types of incorrect maps)
   a. User should be notified if the selected map file is not correct

Game Play

1. Implementation of a game driver implementing the game phases according to the Risk rules.

Startup Phase

1. Game starts by user selection of a user-saved map file.
    a. User should be able to select a valid map to play

2. Map is loaded as a connected graph, which is rendered effectively to the user to enable efficient play
    a. Render user allocated countries as lists

3. User chooses the number of players, then all countries are randomly assigned to players.
    a. User should be able to select the number of players and name them
    b. Countries are randomly allocated to each user

4. Players are allocated a number of initial armies, depending on the number of players
    a. Users should be allocated initial set of armies based on the number of users

5. In round-robin fashion, the players place their given armies one by one on their own countries
    a. UI driven army allocation to User's country based on their selection
    b. At most, a country can have 12 armies. Surplus armies are discarded.

Reinforcement phase

1. Calculation of correct number of reinforcement armies according to the Risk rules
    a. In turn, player should be given additional armies
    b. Maximum of 3 or number of player owned countries divided by 3 is allocated to the user
    c. If a player owns a continent, additional armies are given based on the number of armies

2. Player place all reinforcement armies on the map
    a. Player should be able to select a country and place the required number of armies on his countries
    b. Reinforcement does not end till player has exhausted all additional armies or if all countries exceed the maximum allocation limit of 3. In case the limit is exceeded, surplus armies are discarded

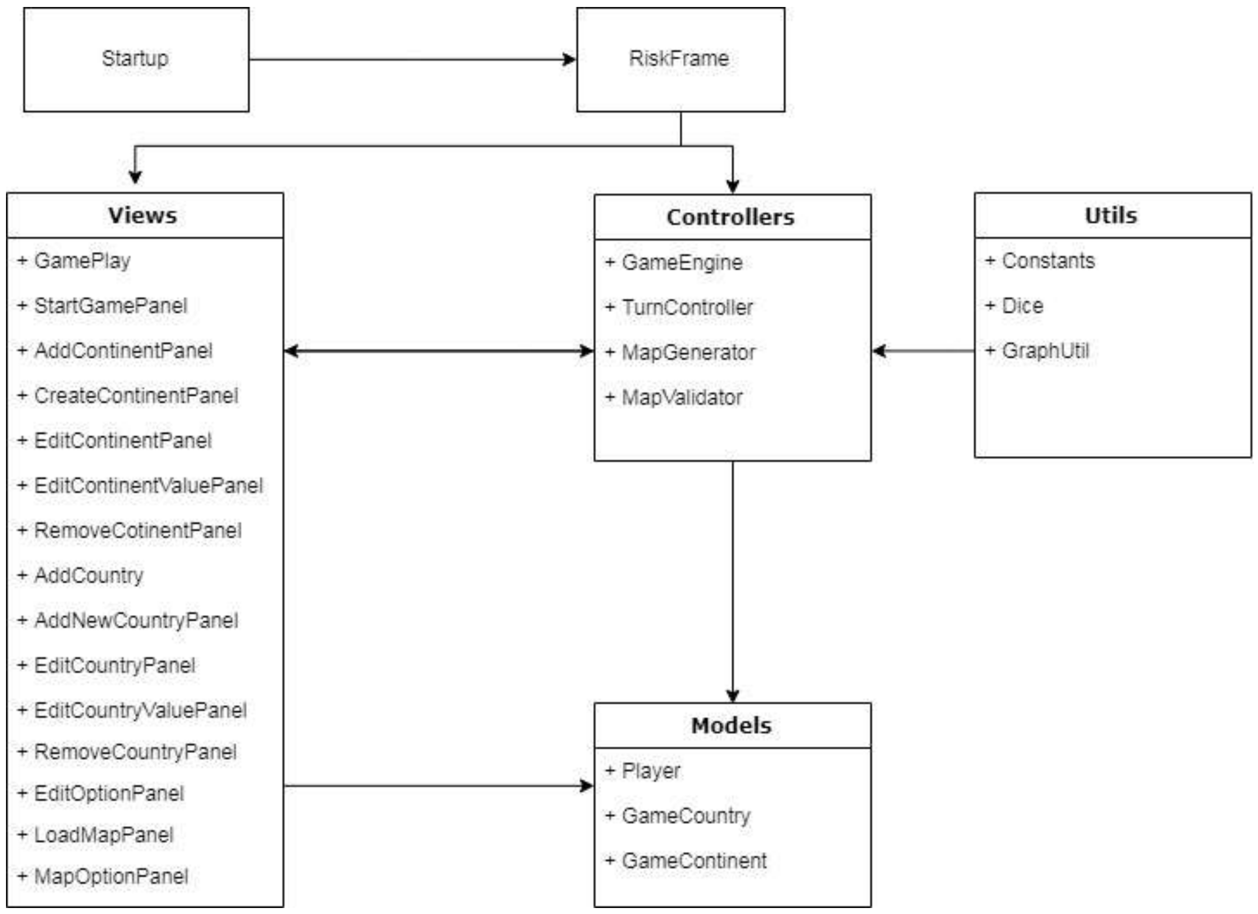
<u>Fortification phase</u>

1. Implementation of a valid fortification move according to the Risk rules
    a. Player should be able to move armies between the player owned countries if the countries are immediate neighbors and the receiving country does not exceed the maximum allocation limit
    b. Player can only move armies between one set of countries in each turn

Architecture

The application follows a simple Model-View-Controller (MVC) architecture. This helps in designing and implementation of cohesive and loosely coupled modules.

Diagram



1. **<u>Models</u>**

These classes represent the basic entities required for the game.

Player	Model which contains player related data
GameCountry	Represents the Country objects used in the Game
GameContinet	Represents the Continent Model in the game

2. **<u>Controllers</u>**

These classes handle the interaction between the models and the views

GameEngine	Controller class to maintain the game state and handle object allocation for the game
TurnController	Controller for handling turn based game logic
MapGenerator	Controller for reading and initializing the game map
MapValidator	Controller containing validations for the generated map

3. **<u>Utils:</u>**

These are utility classes to be used in the game

Dice	Utility to simulate a dice and roll it
Constants	Provides the basic constants values for game
GraphUtil	Utility which uses JGraphT library for graph operations

4. **<u>Views</u>**

These classes are responsible for rendering the UI of the game

RiskFrame	Initial View
StartGamePanel	View for loading initial game details
GamePlay	View for main game play
AddContinentPanel	View to add a new continent into the map after reading map
CreateContinentPanel	View used to add a new continent in the map in creating new map
EditContinentPanel	View for redirection to create, remove or edit the continent
EditContinentValuePanel	View for edit functionality for the existing continent
RemoveContinentPanel	View for the removal procedures for the continent
AddCountry	View for for adding a new country in creating new map
AddNewCountryPanel	View to add the country to an existing map

EditCountryPanel	View to show all the options for manipulating country information
EditCountryValuePanel	View to edit an existing country in an existing map
RemoveCountryPanel	View to remove country from an existing map
EditOptionPanel	View redirect the flow to either editing country or continent
LoadMapPanel	View for read, write and creation of map file
MapOptionPanel	View to redirect flow to load existing map file