Concordia University

Department of Computer Science and Software Engineering

Advanced Programming Practices

SOEN 6441 - Fall 2023

Group Name: W4
Submitted to: Prof. Amin Ranj Bar

Members:
Sneh Satishkumar Patel - 40264053
Romit Manishkumar Patel - 40273551
Yash Subhashbhai Chhelaiya - 40257682
Rushi Hasubhai Donga - 40269583
Abdul Sameer Jani Syed - 40272408
Mayank Nilkanth Parmar - 40269385

Risk Computer Game

A developed program which is compatible with the rules and map files and the command-line play of the “Warzone” version of Risk, which can be found at: https://www.warzone.com/.

About Risk Game

- A Warzone game setup consists of a connected graph map representing a world map, where each node is a country and each edge represents adjacency between countries.
- Two or more players can play by placing armies on countries they own, from which they can attack adjacent countries to conquer them.
- The objective of the game is to conquer all countries on the map.

Goal of Build 1

- RiskGame Build 1's goal is to build the basic structure and key functionality of the Java-based Risk game while keeping to the rules and map files of the "Warzone" edition of Risk.
- This initial build specifies the project's folder structure, coding rules, and design principles.
- It also covers game-critical elements like as map development, player management, reinforcement, and deployment.
- Build 1 serves as the foundation for future features, unit tests, and continuous integration/continuous deployment (CI/CD) pipelines.

CODING STANDARDS

Code Layout

- **Indentation:**
  - Used 4 spaces for indentation.
  - The body of loop and conditional statements are indented in relation to their starting lines.
  - Body of a function is appropriately indented beneath its header.
- **Line Spacing:**
  - Blank lines serve the purpose of creating separation within code.
  - They are inserted between class declarations, methods, and significant segments within complex functions to enhance code readability.
- **Positioning of curly braces:**
  - The position of curly braces involves placing an opening curly brace immediately after the statement preceding it, thereby reducing the length of the code.
- **Spacing around operators and operands:**
  - Inserting spaces before and after operators to enhance the legibility of code.

MapHelper.java

```java
/**
 * Helper method to add/remove a new continent to the game map.
 *
 * @param p_mapFileName Name of map file.
 * @return Edited game map
 */

public GameMap editMap(String p_mapFileName) {
    String l_filePath = Constant.MAP_PATH + p_mapFileName;
    this.d_gameMap = new GameMap(p_mapFileName);
    File l_file = new File(l_filePath);

    if (l_file.exists()) {
        System.out.println(p_mapFileName + " map file exists. You can edit it.");
        this.readMap(l_filePath);
    } else {
        System.out.println(p_mapFileName + " does not exist.");
        System.out.println("Creating a new Map named: " + p_mapFileName);

        this.d_gameMap = new GameMap(p_mapFileName);
    }
    return this.d_gameMap;
}
```

Guidelines for Naming

- **Classes:**
  - Class names follow the convention of being written in UpperCamelCase, also known as Pascal Case.
  - Example : class MyClassName { }
- **Method Parameters, Member functions and Data Members:**
  - These are written in lowerCamelCase.
  - For method parameters, we use **‘p_’** as prefix and for data members, we use ‘**d_’** as prefix.
  - Example : int d_mapIndex = 1, public GameMap loadMap(String p_mapFileName)
- **Local Variables:**
  - Adheres to lower camelCase with the prefix **‘l_’** to indicate that it is a local variable.
  - Example : String l_line;
- **Constants:**
  - Uppercase letters with underscores.
  - Example : public static final String MAP_PATH = "src/main/resource/map/";

Guidelines for Writing Comments

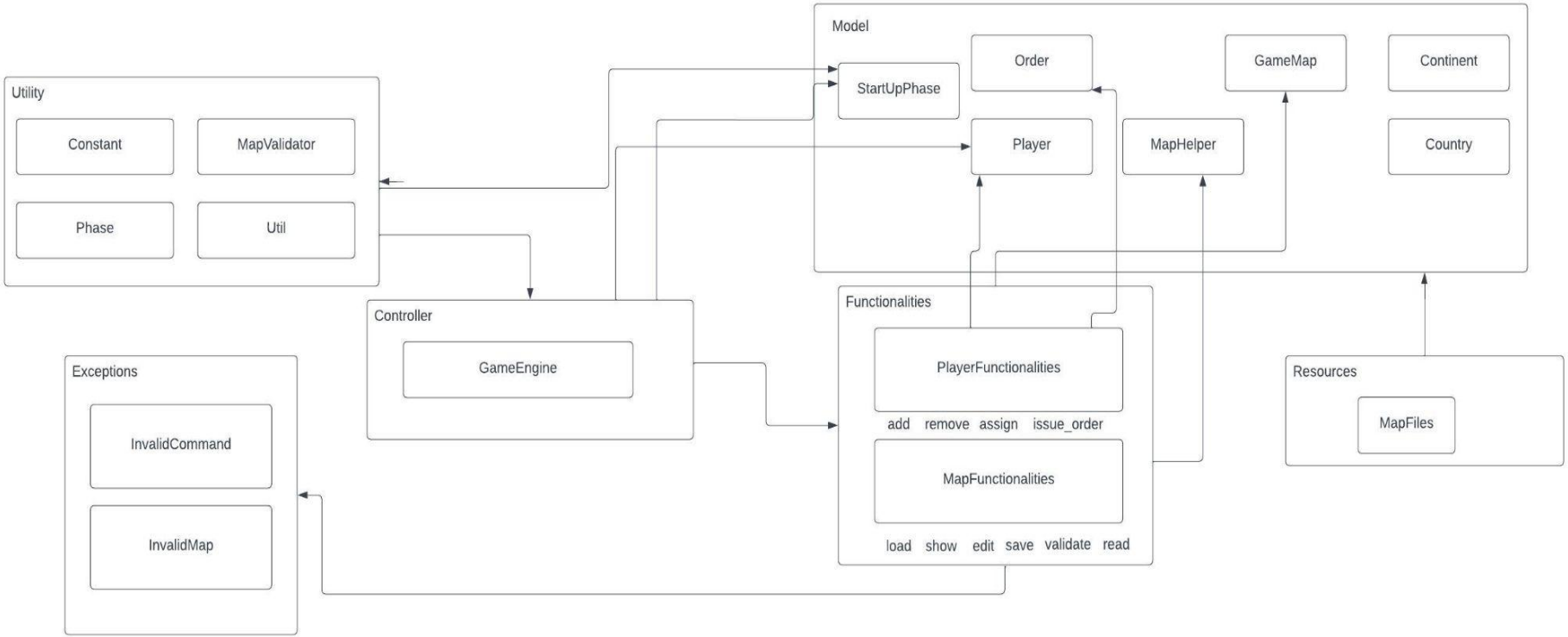
- **Code without any commented-out sections:**
  - Excess commented code is removed to enhance clarity and readability.
- **Documentation comments (JavaDoc):**
  - For each class, data member, and member function, there will be JavaDoc documentation provided above them.
- **In Complex Methods :**
  - In lengthy and complex methods, explanatory content is inserted to improve code comprehension.

StartUpPhase.java

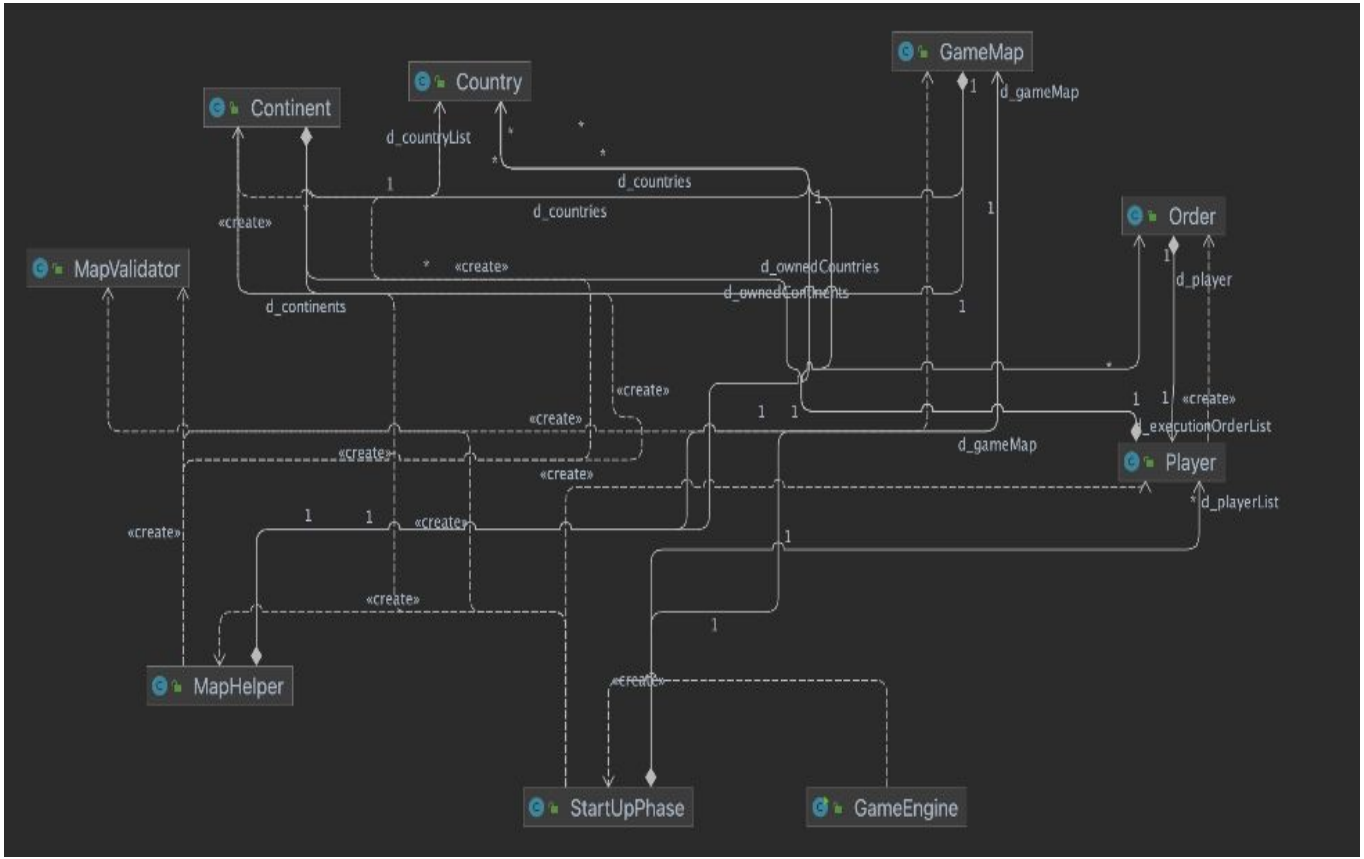
```java
/**
 * Getter method for player list.
 *
 * @return Returns list of game player.
 */
public ArrayList<Player> getPlayerList() {
    return this.d_playerList;
}

/**
 * Setter method for game phase.
 *
 * @param p_gamePhase GamePhase to set
 */
public void setGamePhase(Phase p_gamePhase) {
    this.d_gamePhase = p_gamePhase;
}
```

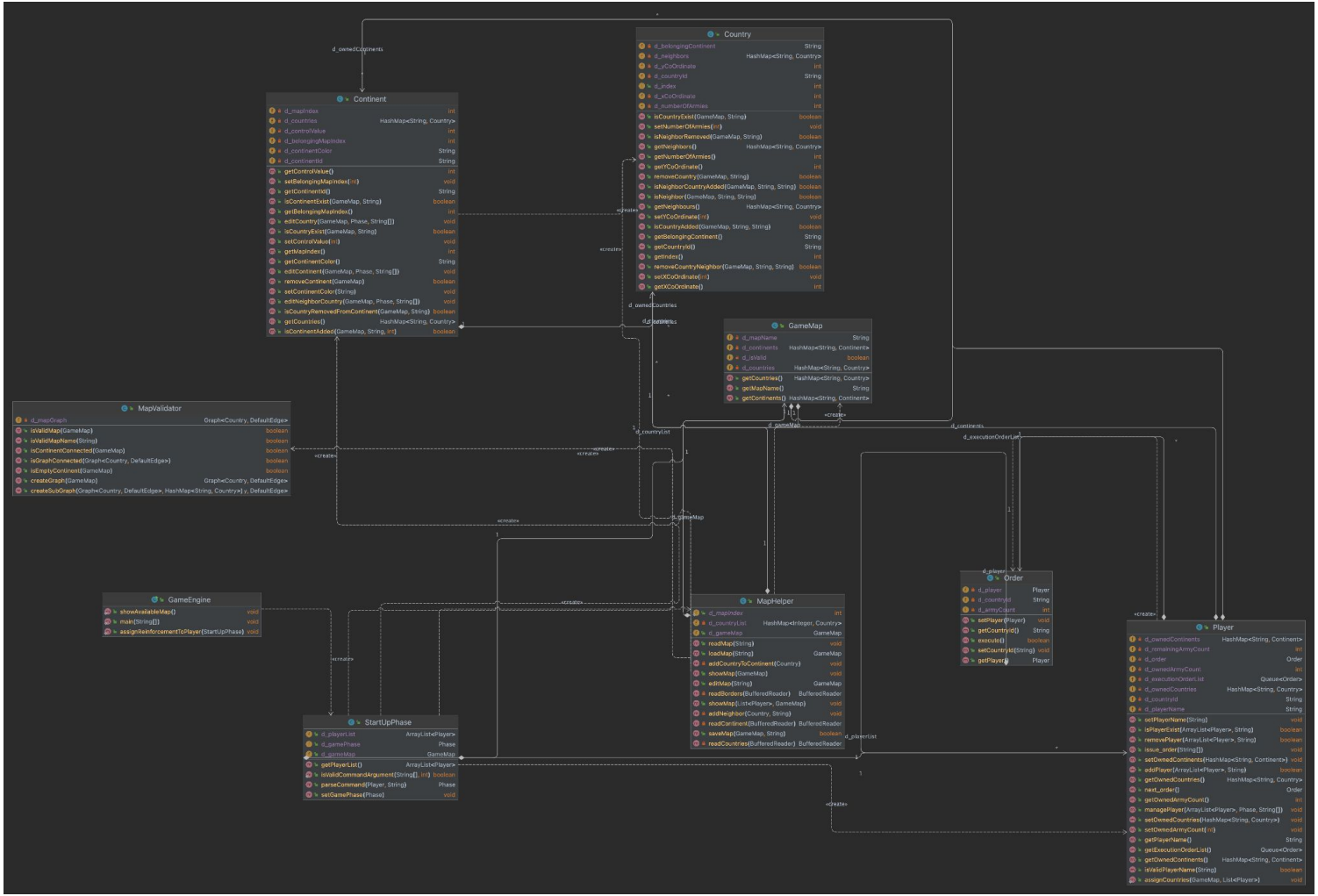
Architecture Diagram



Class Diagram



Functional Diagram


Country
Continent
GameMap
MapValidator
GameEngine
MapHelper
Order
Player
StartUpPhase
«create»
d_ownedContinents
d_gameMap
d_countryList
d_executionOrderList
d_player

References

Warzone game

Domination game maps

As specified in the Project Build 1 document

Thank You