CONCORDIA UNIVERSITY
DEPARTMENT OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

SOEN 6441: Advanced Programming Practices
Fall 2019

Project
Risk Domination Game
(Build 2)

Architectural Design Document

Team Name: Group_U_I

Name	**ID**
Van Tuan Tran	40124288
Benjamin Osei Asante	40080998
Tejinder Singh	40114377
Bharti Saini	40089008
Roger Madhu	40076461

Refactoring

As we are developing the game in iterative development model. Refactoring was done on the basis of notes documented during the demonstration of build 1. Some other refactoring issues was also identified during the analysis of build 2 in our weekly team meeting, which was to minimize copulation between classes and increase modularity. Some of the major refactoring which has taken place are as follows:

1. Overall system structure (Folder organization)
2. Command line input object model creation
3. Implementation of better exception handling methods
4. Separation of objects
5. Improved validation for maps
6. Improved map creation model

Overall system structure (Folder organization)

In respect to build 1, our folder structure was organized more separately, for example, the folder "commands" containing the files "GameCommands.java" and "MapEditorCommands.java" was in separate folder which has been taken inside the "Controllers" folder due to the corelation of tasks between the two.

Command line input object model creation

We have changed the traditional command parser from build 1 to object creation parser. Now all command line inputs are converted into command model object which is fed to our game and the game interprets it as necessary.

Implementation of better exception handling methods

We have reviewed the scope of our game and implemented enhanced exception handling, for example, the inputs are now tested where necessary for integer, double, string, wrong input. If the program finds an error in the input or the game, it handles it by separated handler for NullPointerException, NumberFormatException, InvalidInputException, etc.

Separation of objects

The singleton object GameBoard has been separated into GameBoardMap, GameBoardPlayer, Resettable. The Country, Player, Continent has also been refactored as was necessary to segregate classes and improve modularity.

Improved validation for maps

The map tree validation methods has been improved as noted in build 1 demo. Now the map validator contains additional validation criteria to ensure proper working of the system.

Improved map creation model

The map creation model which generates the map file has been improved as per notes from build 1 demo. Additional information has been removed from the map file.

All the refactoring was done on files which are associated with test cases. We also have automatic test suite setup in our github repository ensuring every push does not break the codebase.