



RISK GAME (BUILD-1)

ARCHITECTURAL DESIGN

Advanced Programming Practices

SOEN 6441
Fall-2019

Team: Group_U_J

Mehul Prajapati	(40076930)
Komal Panchal	(40130791)
Maryam Giahi	(40016260)
Mahmoudreza Entezami	(40058782)
Parth Patel	(40081116)
Shubham Ranadive	(40083991)

Contents

<i>Introduction</i>	3
1. SCOPE	3
1.1 MAP EDITOR:	3
1.2 GAME PLAY:	3
2. <i>Architecture Design</i>	4
3. <i>UML DIAGRAM</i>	5
4. Modules Description	6
4.1. Config	7
4.2. Entity	7
4.3. Mapper	7
4.4 Main	8
5. Test Cases (JUnit) Description	8
5.1 Main Test	8
5.2 Entity Test	8
5.3 Mapper Test	8
6. Tools and API	9
7. References:	10

Introduction

Developed an operational version demonstrating a subset of the capacity of Risk game. It was an effort to implement several extreme programming key features such as Collective ownership, Coding Standards, JavaDoc, Unit Testing and many more.

1. SCOPE

The scope of the build 1 is as per the instruction guidelines for the build:

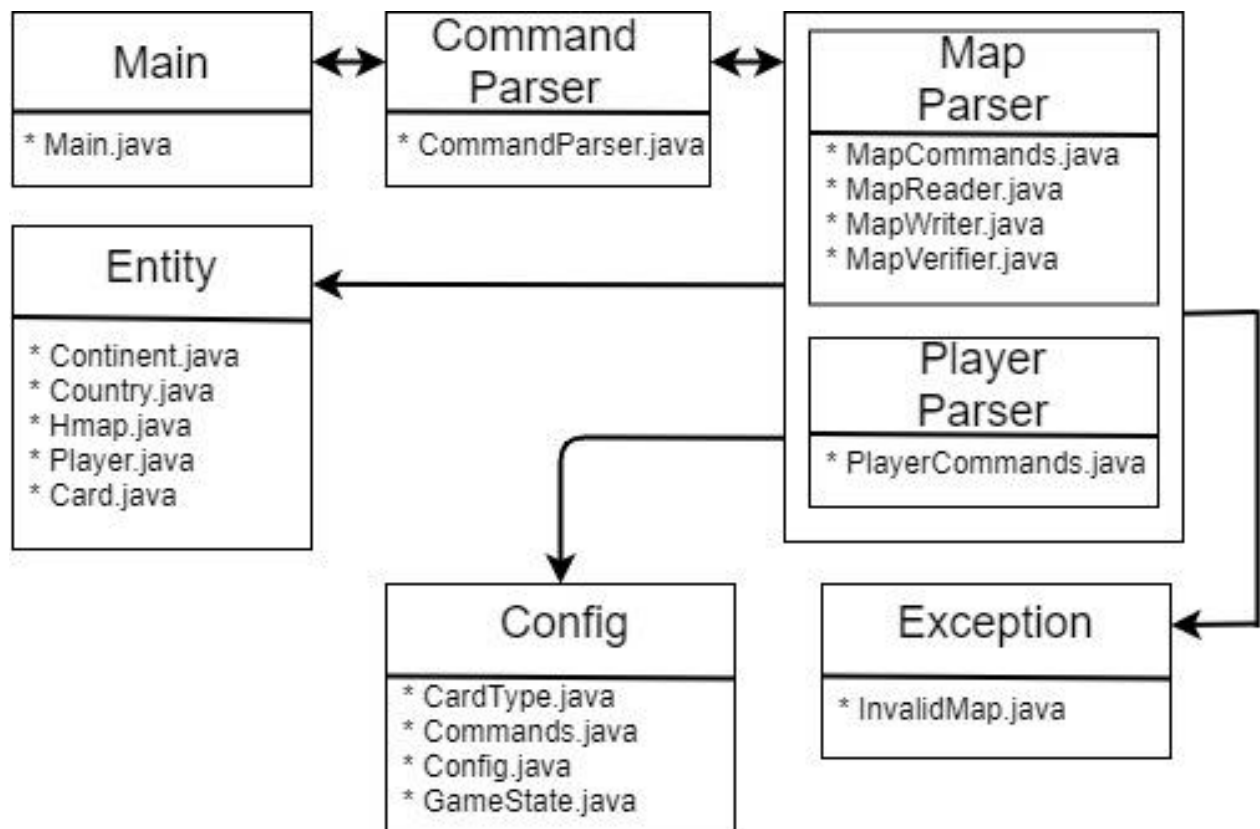
1.1 MAP EDITOR:

- Create a new map file
- Edit an existing map file
- Add/Update/Delete Continent, Country and Adjacent Country
- Make sure that the integrity of the connected graph is maintained.

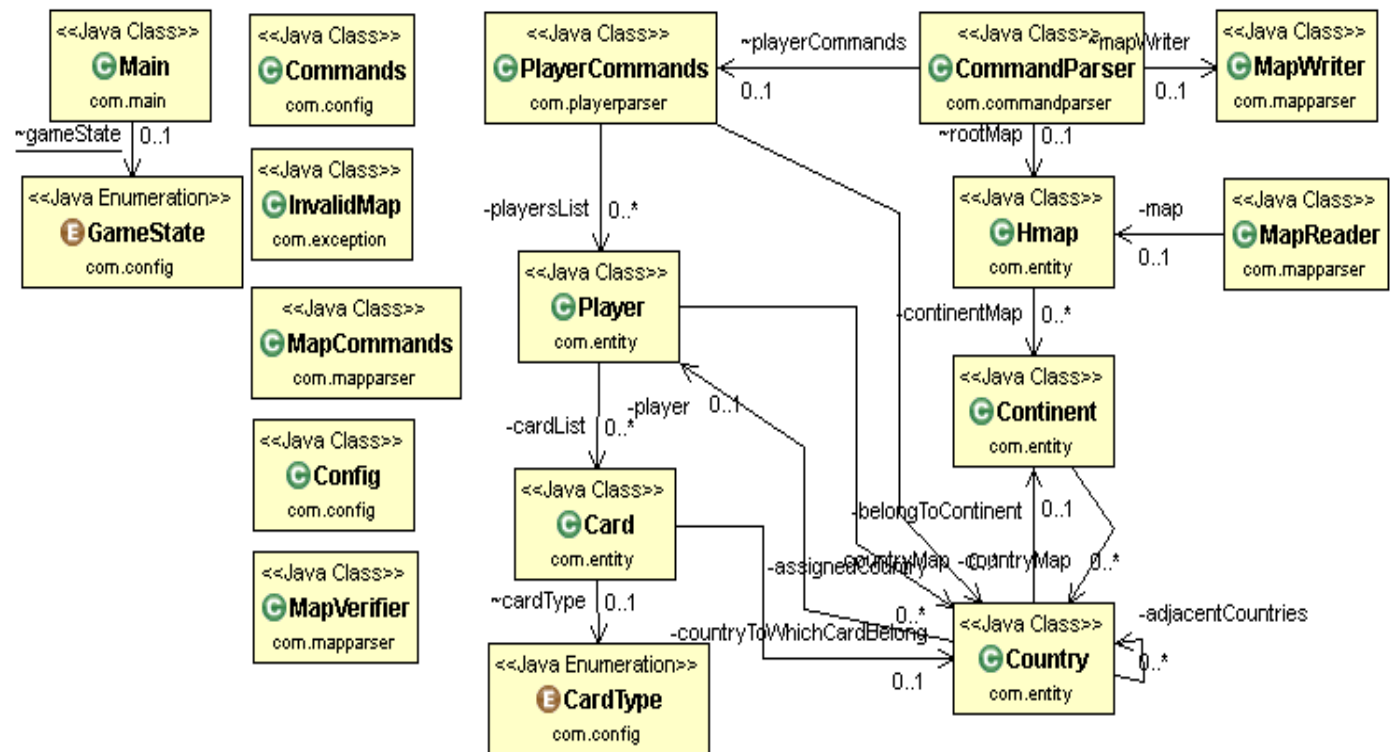
1.2 GAME PLAY:

- Assigning country to player
- Player can assign armies to each country in round robin manner
- With proper calculation of armies, Reinforcement phase is implemented
- With a valid fortification move, Fortification phase is implemented

2. Architecture Design



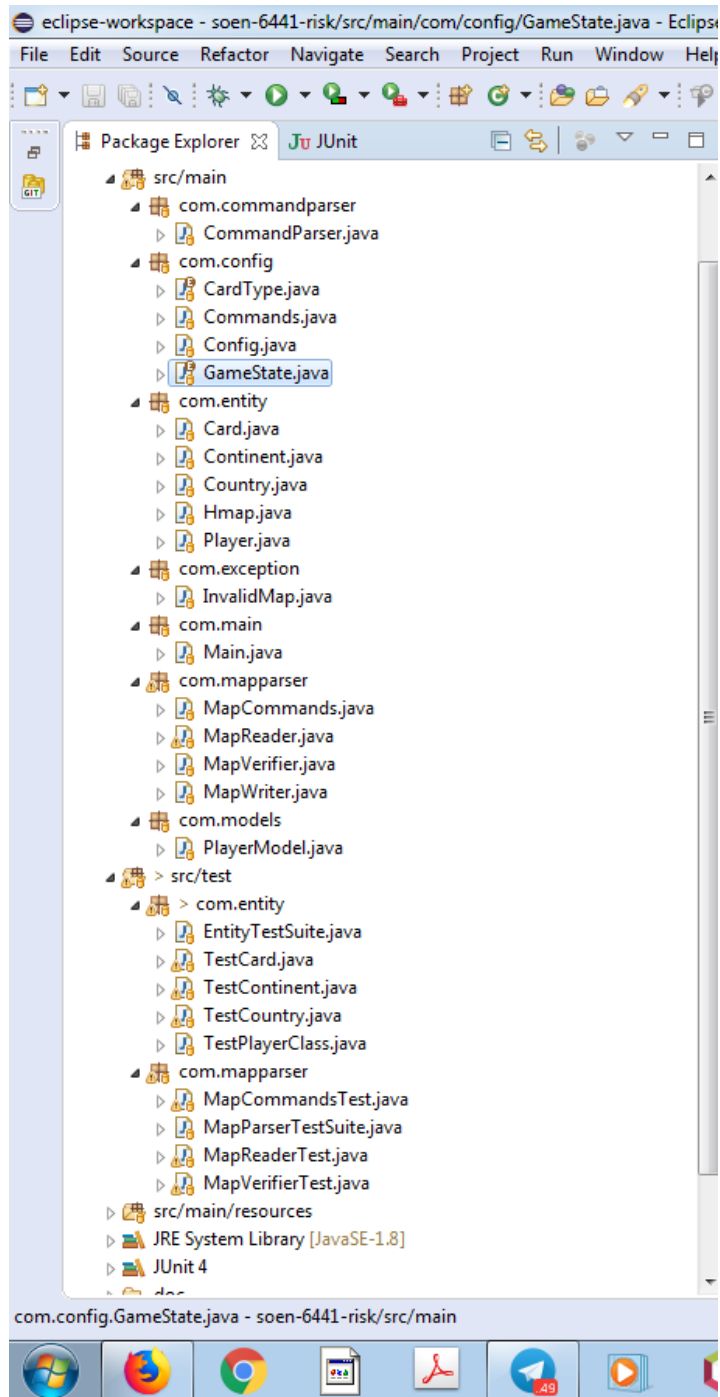
3. UML DIAGRAM



4. Modules Description

Detailed Structure and Explanation

The overview of project packages is as shown in Screenshot 1.



4.1. Config

File_name	Description
CardType.java	It is a class for cardtype configuration.
Commands.java	It is a class for Command configuration.
Config.java	It is a class for army configuration .
GameState.java	It is a class for GameState configuration. .

4.2. Entity

File_name	Description
<i>Hmap.java</i>	It contains all the information of the Map and a list of the continents.
<i>Card.java</i>	It contains all the information regarding to Card.
<i>Continent.java</i>	It contains all the information of the continent and a list of all the countries that belong to a continent.
<i>Country.java</i>	It contains the information of the country like name, a reference to which continent the country belongs, list of all the adjacent country, count of armies currently residing on the country.
<i>Player.java</i>	It contains all information related to a player and the number of armies assigned to the player.

4.3. Mapper

File_name	Description
MapCommand.java	Contains all the common method of the map like: saving map
MapReader.java	It reads the map file format and parsing in to Map object and checks for the validity of data of the map file.
MapWriter.java	It is responsible for writing the Map object to the file.
MapVerifier.java	object, opening a dialogue box etc.

4.4 Main

File_name	Description
Main.java	Entry point for the application

5. Test Cases (Junit) Description

5.1 Main Test

File_name	Description
MainTestSuite.java	This is a test class for running all test suits. (MapUtilitiesTestSuite.class, GameUtilitiesTestSuite.class, ModelsTestSuite.class, StrategyTestSuite.class)

5.2 Entity Test

File_name	Description
TestCard.java	This is a test for card class.
TestContinet.java	This is a test for continent class.
TestCountry.java	This is a test for country class.
TestPlayer.java	This is a test for player class.
EntityTestSuite.java	This is a test class for running all test suits in Entity i.e. (TestCard.class, TestContinent.class, TestCountry.class, TestPlayerClass.class)

5.3 Mapperparser Test

File_name	Description
MapcommadTest.java	This is a test class for Map operations
Mapreader.java	This is a test class for Map-reader.
Mapwriter.java	This is a test class for Map writer.
Mapverifier.java	This is a test class for Map verifier.

EntityTestSuite.java	This is a test class for running all test suits in Mapparser i.e. (MapcommadTest.class, MapreaderTest.class, MapwriterTest.class, MapverifierTest.class)
----------------------	--

6. Tools and API

<i>Tools</i>	<i>Description</i>
Eclipse	IDE for the game development
Git	It is Git code management System which gives one place to plan projects, collaborate on code test and deploy.
Junit4	For writing test class
JavaDoc	JavaDoc is a automation tool to generate document to refer regarding libraries and modules of code.

7. References:

1. Rules Followed : <https://www.wikihow.com/Play-Risk>
2. <https://sourcemaking.com/refactoring/refactorings>
3. <https://www.sourcetreeapp.com>