System design document for ”Chalmers Risk”

Version: 3.00

Date: 26/5 - 2015

Author: Malin Thelin, Oskar Rutqvist, Björn Bergqvist, Robin Jansson

This version overrides all previous versions.

1 Introduction

1.1 Design goals

To support testing the software must support the standards of object oriented programming, it must be possible to test isolated part of the program.

1.2 Definitions, acronyms and abbreviations

- GUI, Graphical User Interface.
- Java, the programming language used to produce this application.
- JRE, Java Runtime Environment. This is necessary to be able to run the application. Java applications are run via this platform.
- MVC, Model - View - Controller. Design pattern to be used to avoid mixing up model and controller code together as well as keeping the interface apart from these parts.
- Territory, the smallest are of the game to be occupied.
- Continent, a “collection” of territories.
- Troop, used to conquer and defend territories.
- Turn, The application is a turn based game with two human players and thus each player will take turns performing his/her actions.
- Card, event cards and will be awarded to players who conquered a territory on their previous turn. Event cards will impact the game in various ways, granting troops to the player or ending the game prematurely.

2 System design

2.1.1 Overview

The application will follow the MVC design pattern. Our implementation of MVC might be a little different than the ordinary MVC model. Our goal is to make the design as simple as possible. We want the interface to solemnly show the functions that is available for the player at the current point of time.

2.1.X Rules ******SHOULD REMOVE******

Three phases of a turn: deployment, attack and move. If playing with cards a draw card phase is going to happen prior to the deployment phase.

Deployment - Count the number of territories and continents to see how many troops a player gets. Then the player get to choose which territory he or she wants to place said troops in.

Attack - A player selects a territory and an amount of troops, not greater than one less than the current troops receding in said territory, to attack with and then moves them to an adjacent territory owned by another player.

Move - A player selects a territory and an amount of troops, not greater than one less than the current troops receding in said territory, the player can then select a second territory he controls that are connected to the first one in order to move the selected troops there.

Card - Event cards are awarded to players who on their previous turn has conquered a territory and the cards can impact the game in various ways.

2.2 Software decomposition

2.2.1 General

Package diagram. For each package an UML class diagram in appendix.

2.2.2 Decomposition into subsystems

2.2.3 Layering

See appendix (NYI)

2.2.4 Dependency analysis

See appendix (NYI)

2.3 Concurrency issues

There are no concurrency issues, the game is a single threaded application.

2.4 Persistent data management

NA - The application does not implement any method for saving games between sessions.

2.5 Access control and security

NA

2.6 Boundary conditions

NA - The application is launched and exited as a normal java file.

3 References

MVC: http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
RISK: http://en.wikipedia.org/wiki/Risk_(game)

APPENDIX

- INSERT USE-CASES HERE. TABLES AND TEXT.