Requirements and Analysis Document for Group 25

Version: 4

Date: 2015-05-26

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

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The purpose is to create a computer game based on the classic board game Risk. However, we will change it so that instead of taking over the world, the player(s) will take over Chalmers.

1.2 General characteristics of application

The game is going to become a multiplayer desktop application, but the game will not use any networking and the players are assumed to sit around the same computer.

The application will be a turn based strategy game and will mostly follow the traditional rules of Risk. The game will be a turn based application with the order of the players randomized. The game will handle player points, troops handouts and calculating combat results, event cards will be given to player who conqured a territory on the previous turn.

1.3 Scope of application

We intend to implement the game as a two-player game. The idea is the two players can sit at the same computer and play against each other to conquer the world of Chalmers. The game won't be playable alone unless the player decides to play against her-/himself.

1.4 Objectives and success criteria of the project

The project is deemed to be finished when we have a fully implemented Risk game. Two players should be able to play the game on the same computer, taking turns. One player wins when the other is out of troops and territories.

1.5 Definitions, acronyms and abbreviations

Turn based: If there are two players (or more) everyone gets their own turn to do objectives. While one player has their turn, other players are unable to do anything.

Board game: The application is a simulation of a tabletop board game.

Java: A programming language that is independent on the platform.

JRE: Java Runtime Environment, additional software required to run our application

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

Create a list of high level functions here (from the use cases).

-
- Start a new game
-
    -
    - select a team (name and a color)
    -
-
- Place your armies
-
    -
    - take turns placing forces until there are no more forces to place by either player.
    -
-
- Take a turn
-
    -
    - place your new forces (the forces you received at the beginning of this turn)
    -
    - fight an enemy
    -
    - relocate forces (to adjacent territories)
    -
    - end turn

    -
-
- Win/Lose

2.2 Non-functional requirements

2.2.1 Usability

The interface should be very intuitive, clear and easy to understand. Included with the game is a readme file containing the game rules, and an explanation of the interface.

2.2.2 Reliability
NA

2.2.3 Performance

Max 2 sec response time, worst case. Performance is a minor issue for our project since its a turn based game and not a real-time computer game.

2.2.4 Supportability
The game will be implemented in such a way that it will be easy to use different maps for the “world”.

2.2.5 Implementation
To ensure platform independence the application uses the Java Environment. To play the game on a computer it must therefore have JRE installed and configured, and of course, also the game.

2.2.6 Packaging and installation
NA

2.2.7 Legal
NA

2.3 Application models

2.3.1 Use case model

UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority

1, Start a game
2, Perform combat (roll dice)
3, Move troops
4, Place troops
5, end turn

2.3.3 Domain model

See APPENDIX.

2.3.4 User interface

Text to motivate a picture.

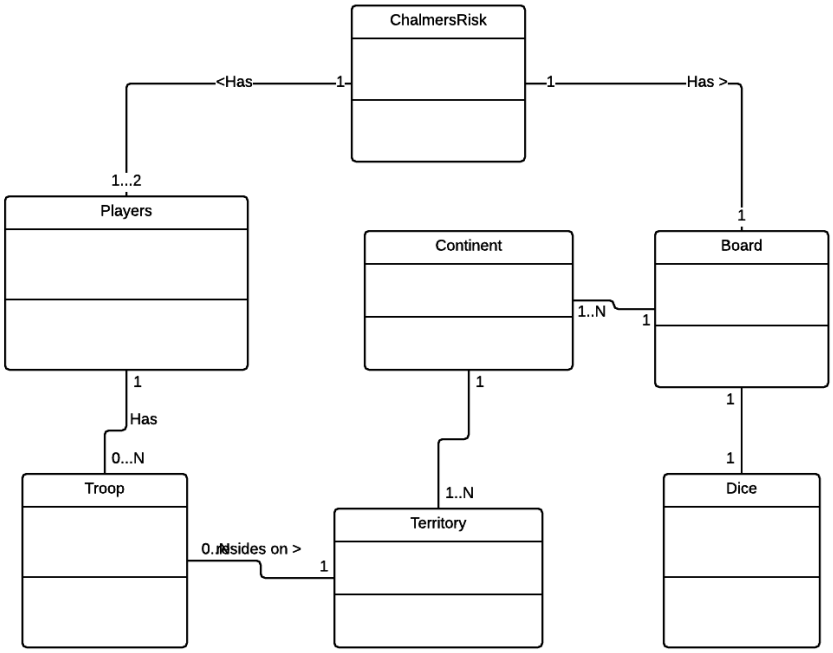
2.4 References

Risk board game: http://en.wikipedia.org/wiki/Risk_%28game%29

APPENDIX

GUI

Domain model



Use case texts