

Requirements and Analysis Document for the Chalmers Risk Project (RAD)

Contents

1. **Introduction**
 - 1.1 Purpose of Application
 - 1.2 General characteristics of application
 - 1.3 Scope of Application
 - 1.4 Objectives and success criteria of the project
 - 1.5 Definitions, acronyms and abbreviations
2. **Requirements**
 - 2.1 Functional requirements
 - 2.2 Non-functional requirements
 - 2.2.1 Usability
 - 2.2.2 Reliability
 - 2.2.3 Performance
 - 2.2.4 Supportability
 - 2.2.5 Implementation
 - 2.2.6 Packaging and installation
 - 2.2.7 Legal
 - 2.3 Application models
 - 2.3.1 Use case model
 - 2.3.2 Use cases priority
 - 2.3.3 Domain model
 - 2.3.4 User interface
 - 2.4 References

Version: 5

Date: 2015-05-31

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

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 conquered a territory on the previous turn.

1.3 Scope of application

We intend to implement the game as a two-player game. The two players will 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 player no longer controls any 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

Two players should be able to:

1. Start a game of Chalmers Risk
2. Recieve and place troops on the board
3. Go to the next phase
4. End their turn, giving the next player the control
5. Attack another players territory and either:
 - Win and take over that territory
 - Lose and the defender gets to keep their territory
6. Move troops between their own territories, as long as they are connected.
7. Win the game by taking over all of the territories
8. Exit the game at any time

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

See APPENDIX

2.3.2 Use cases priority

1. Start a game
2. Place troops
3. End turn
4. Go to next phase
5. Perform combat (roll dice)
6. Move troops

2.3.3 Domain model

See APPENDIX.

2.3.4 User interface

User interface is intuitive and text shows up telling the user what to do.
See APPENDIX for pictures.

2.4 References

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

APPENDIX

GUI

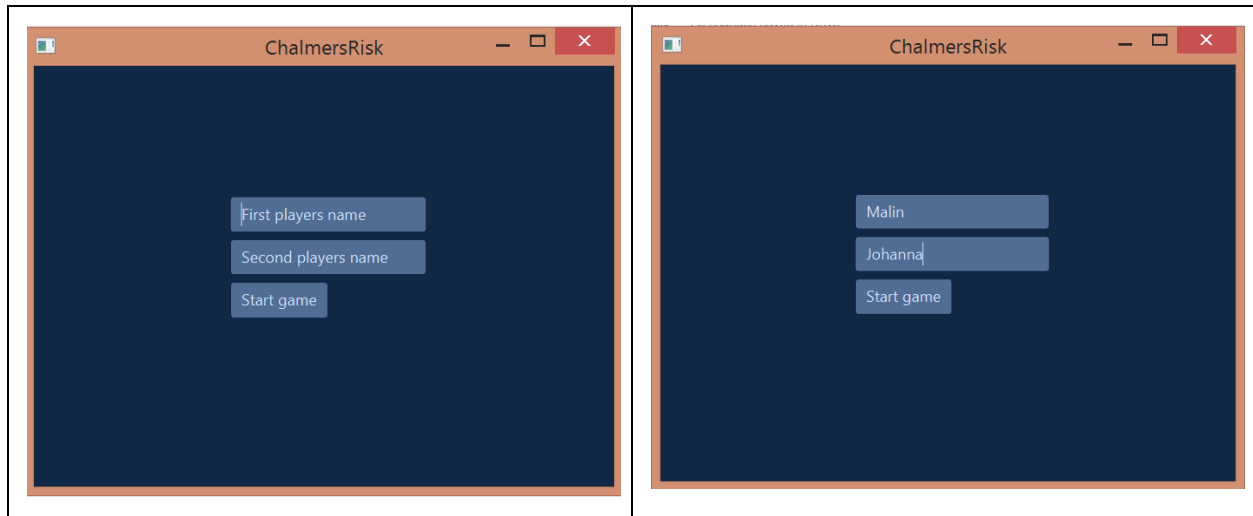


Figure 1: The startscreen that comes up when you run the application.

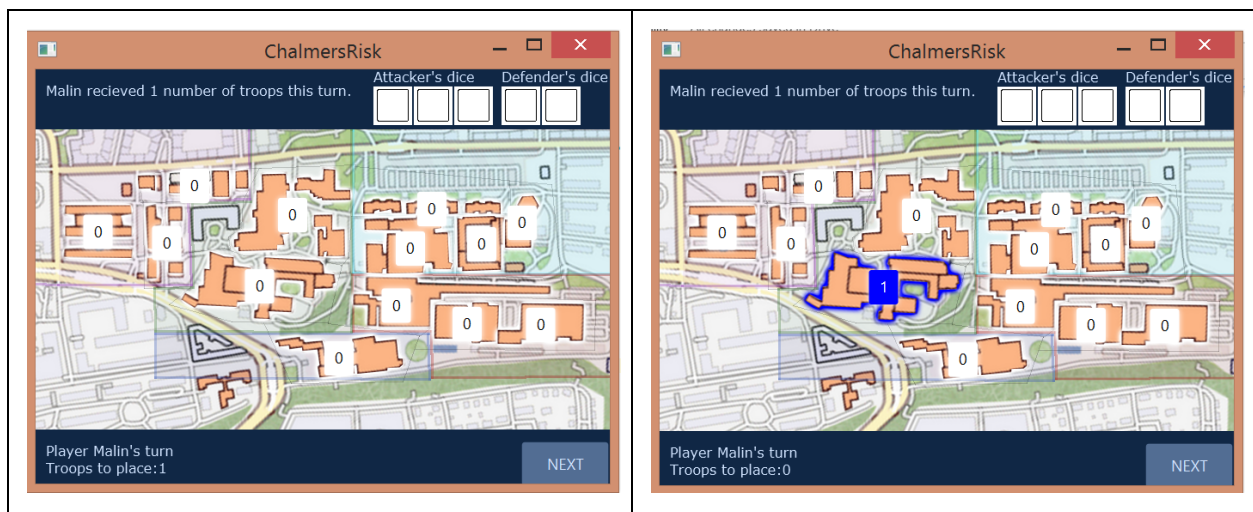


Figure 2: The gameboard in phase 1 - Place Troops. Showing before and after Player Malin has placed their first troop.



Figure 3 : The gameboard before and after getting to phase 2 - Attack phase.

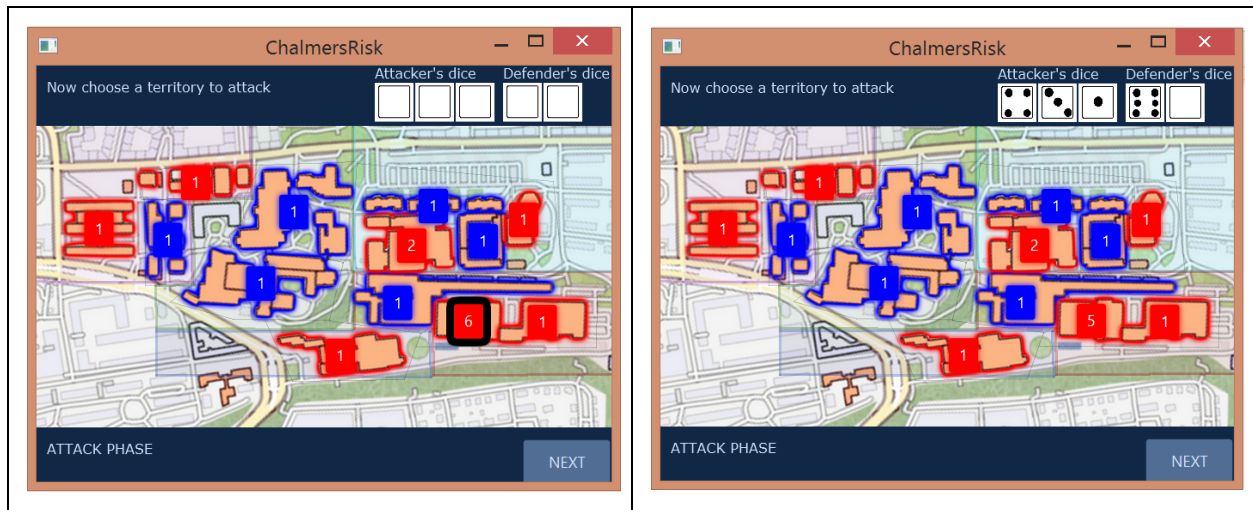


Figure 4 : The gameboard before and after an attack, that in this instance did not end in victory for the attacker.

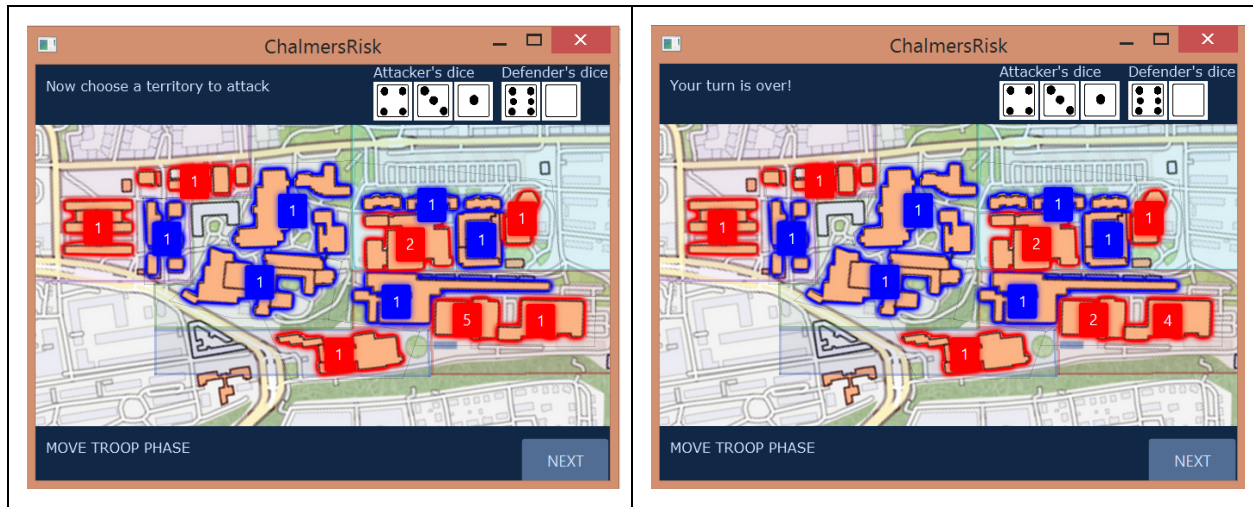
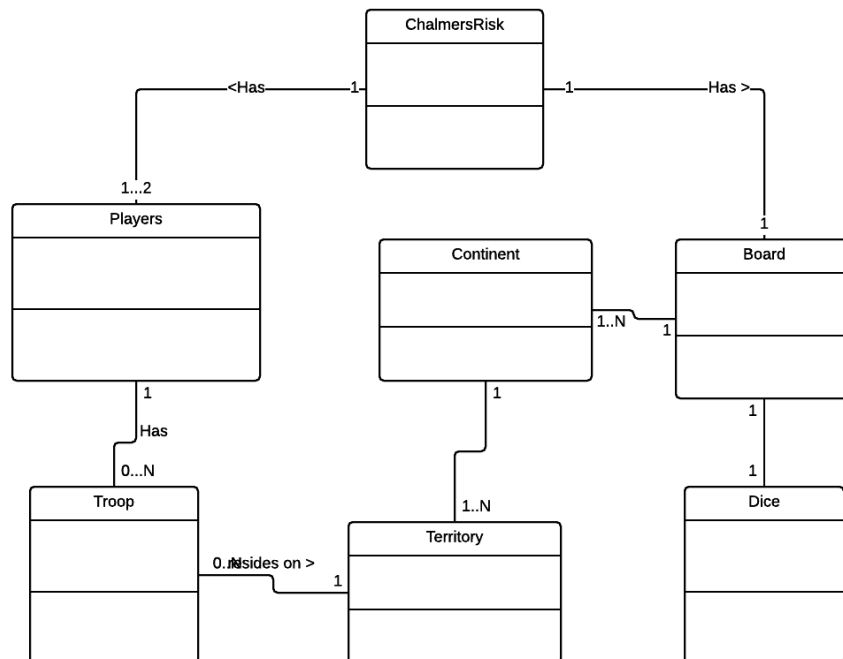
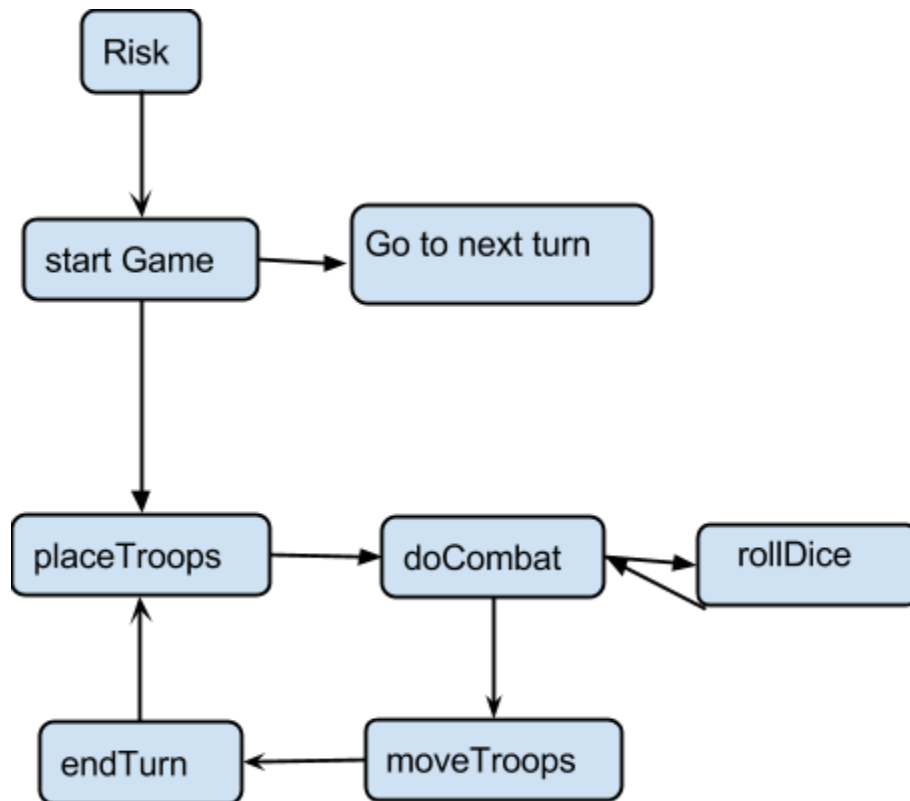


Figure 5 : The gameboard before and after the red player moved three troops from one territory to another.

Domain model



Use case overview:



Use case texts:

Use Case : Start a game

Summary

The players will start a game of Chalmers Risk.

Priority

High

Extends

-

Includes

Participators

Players

Normal flow of events

User:	System:
1. Enters the names of the players	
2. Presses "Start game" button	
	3. Initializes the game.

Alternate flow of events

User:	System:
1. Presses "Start game" button	
	2. Requests that you enter names before

	starting the game.
--	--------------------

Use Case : Roll Dice

Summary

Rolls dice to determine who wins at combat

Priority

Medium

Extends

Perform combat

Includes

Participators

Attacking territory, defending territory

Normal flow of events:

User:	System:
1. UC: Perform Combat	
	<p>2. Depending on the amount of troops the attacking territory has; chooses the amount of dice.</p> <p>1 if the amount is 1. 2 if the amount is 2. 3 if the amount is 3 or higher.</p> <p>Depending on the amount of troops the defending territory has; chooses the amount of dice.</p> <p>1 if the amount is 1. 2 if the amount is 2 or higher.</p>
	<p>3. Rolls the dice and compares the values. Remove attacking player's and defending player's troops, depending on how it went.</p>

Use Case : Perform combat

Summary

A player will attack another player's territory.

Priority

Medium

Extends

-

Includes

Participators

Attacking player, defending player

Normal flow of events: 1a

The attacking player attacks another territory and wins.

User:	System:
1a. Chooses a territory that they own	
	2a. Marks the territory as chosen
3a. Chooses a territory that they do not own	
	4a. UC Roll Dice.
5a. Attacking player wins	
	6a. Sets attacking player as the new owner of defending player's territory. Moves all of attacking player's troops -1 to their new territory.

Normal flow of events: 1b

The attacking player attacks another territory and loses or it's a draw.

User:	System:
5b. Attacking player loses or it's a draw	
	6b. Defending player keeps their territory.

Alternate flow of events

The attacking player tries to choose a territory that the do not own.

User:	System:
1b. Chooses a territory that they do not own	
	2b. Does nothing.

Alternate flow of events

The attacking player tries to attack one of their own territories.

User:	System:
3b. Chooses a territory that they own	
	4b. Does nothing

Use Case : Move Troops

Summary

A player will move a chosen amount of troops from one territory to another.

Priority

Medium

Extends

-

Includes

Participators

Current player

Normal flow of events: 1a

User:	System:
1. Clicks a territory they own with more than 1 troop on it.	
	2. Tells the user they have chosen that territory and that they currently have chosen to move one troop from there.
3a. Chooses more troops to move by clicking the same territory.	
	4a. For each click tells the user how many troops they want to move. When there is only one player left the system will not allow the player to choose more troops.
5. Chooses another territory they own	
	6. Removes the chosen amount of troops from the first chosen territory. Adds the the

	troops to the other chosen territory.
--	---------------------------------------

Normal flow of events : 1b

User:	System:
1. Clicks a territory they own with more than 1 troop on it.	
	2. Tells the user they have chosen that territory and that they currently have chosen to move one troop from there.
3b. Chooses another territory they own	
	4b. Removes the chosen amount of troops from the first chosen territory. Adds the the troops to the other chosen territory.

Alternate flow of events:

User:	System:
Either 1a or 1b.	
2. Chooses a territory they do not own.	
	3. Does nothing

Alternate flow of events:

User:	System:
1. Clicks a territory they do not own	
	2. Does nothing

Use Case : Place Troop

Summary

A player will place one troop on a chosen territory.

Priority

High

Extends

-

Includes

Participators

Current player

Normal flow of events: 1a

User:	System:
1a. Chooses an empty territory.	
	2a. Places one troop on the selected territory. Makes the current player the owner of this territory.

Normal flow of events: 1b

User:	System:
1b. Chooses a territory they own	
	2b. Places one more troop on the selected territory.

Alternate flow of events: 1c

User:	System:
1c. Chooses a territory that is owned by another player.	
	2c. Does nothing

Use Case : End Turn

Summary

The current player ends their turn, making it the next player's turn.

Priority

High

Extends

Move troops, Go to next phase

Includes

Participators

Current player, next player

Normal flow of events

User:	System:
	1. Is in the Move Troop phase.
2. Presses "End Turn" button	
	3. Changes the current player to the next player.

Use Case : Go to next phase

Summary

The current player clicks the next button and goes to the next phase in the order.

Priority

High

Extends

Includes

Participators

Current player, Place troop phase, attack phase, move troop phase

Normal flow of events : 1a

User:	System:
	1. Is in the Place Troop phase
2. Clicks the "Next" button	
	3. Goes to Attack phase

Normal flow of events : 1b

User:	System:
	1. Is in the Attack phase
2. Clicks the "Next" button	
	3. Move troop phase

Alternate flow of events : 1c

The player tries to press next when they still have troops to place

User:	System:
	1. Is in the Place Troop phase
2. Clicks the "Next" button	
	3. Alerts the player that they still have troops to place before they can go to the next phase.

Alternate flow of events : 1d

The game is still in the Initial Phase, meaning it can't go to the attack phase.

User:	System:
	1. Is in the Place Troop phase
2. Clicks the "Next" button	
	3.Changes players. Sets the phase to 'Place troop phase'.