Requirements and Analysis Document for Risk project (RAD)

1 Introduction

1.1 Purpose of application

The project's purpose is to create an Android mobile version of the board game Risk. The purpose was also to make a MVC-model separated from the graphics, ui and network etc so we easily could reuse the logic for other devices.

1.2 General characteristics of application

The game is turn based and can be played either on a single device or by multiple players on separate devices. The GUI will be similar to the board game, but with relevant changes to work well as a touch application and with an overlay with general information and actions.

1.3 Scope of application

The application will not save game and does not include a computer player. Network works globally. 2-4 players will be able to participate in a game. The rules will be similar to the board game, but will not include mission cards.

1.4 Objectives and success criteria for the project

- Should be able to be played as the board game “Risk” (except mission cards),
- Application should be uploaded to Google Play Store.
- Both singleplayer and multiplayer should be fully implemented and have the same functionality.

1.5 Definitions, acronyms and abbreviations

GUI - graphical user interface
Board - The game board
Server - The server hosting the game
Client - Clients connecting to the server
Armies - Units in the game
Region - A region that can be conquered
Turn - One game turn
Continents - A group of regions which award extra armies the next turn
Game - One game played from start until only one player remains
War - One player attacks a region controlled by another player
Dice - Rolled to decide who wins a war

2 Requirements

2.1 Functional requirements

The players should (in not logged in menu) be able to:

1. Play offline
2. Log into Google Player account

The players should (in not logged in menu) be able to:

1. Play offline
2. Play against random opponent(s)

3. Invite friends to a game’
    a. Choose friends from a his google account friends
4. See invitations received

The players should (in game) be able to:

1. Pick starting territories
2. Place starting armies
3. Place armies
4. Fight against an opponent's territory
5. Move armies
6. Hand in cards
    a. Choose which cards to turn in

2.2 Non-functional requirements

2.2.1 Usability

User interface and game logic should be understandable for players with little to none previous experience with the game. Application should communicate with easy to understand language to make it possible for users with basic English knowledge to understand.

2.2.2 Reliability

Since the application is made to support many different devices it might be hard to fully test the application on all of them. Although java and all other technologies used are platform independent some bugs triggered by device resolution or similar might not be found. The developers does not have the resources to buy and test the application on multiple devices therefore it is expected that the application will be a bit unstable at release.

Furthermore the application is mostly based on networking and graphics both of which is hard to automatically test. The man hours required for manually, thoroughly, testing were unfortunately not available.

2.2.3 Performance

The application is quite performant. Most actions are performed without any noticeable delay and the application runs fast overall. There is a bottleneck in speed, importing the data takes approximately two seconds but is only done at startup, this is the only noticeable time-consuming action. In the future the result could be cached and which result in a completely seamless experience.

The network data consumption is minimal, it only sends changes to the other players, each change taking less than a hundred bytes.

2.2.4 Supportability

The application is written in Java. The graphic package is for android version 15 and higher although the application is using MVC-model so making it to a web based application should not exceed 2 man-weeks.

The project follows the MVC-model and replacing the GUI, Graphics and network handling should be relatively easy and straightforward thanks to the structure.

Automated tests exist for all use cases. GUI needs to be tested manually.

2.2.5 Implementation
The application needs a Android API15 or newer in it's current state which means that the application works on all devices except for those bough before 2011 that has not since been updated.

According to google play it does support all devices.

2.2.6 Packaging and installation
The application and documentation will be delivered as an zip-archive, containing the following:

1. Apk-file for installation on android units (functioning on Android os 4.0.3 or higher)
2. README file containing information about a multiplayer version of the game

2.2.7 Legal
There are legal issues regarding trademark with the Risk game, there are other popular applications on google play facing the same trademarking issues, so we do not worry.

All images are made inhouse.

2.3 Application models

2.3.1 / 2.3.2 Use case model, Use case priority
Place armies on held territories - High
Attack and conquer a hostile territory - High
Move armies - Medium
Connect to a server - Low
Start game - High
Quit game - Low
Change settings - Low

2.3.3 Domain model
See APPENDIX.

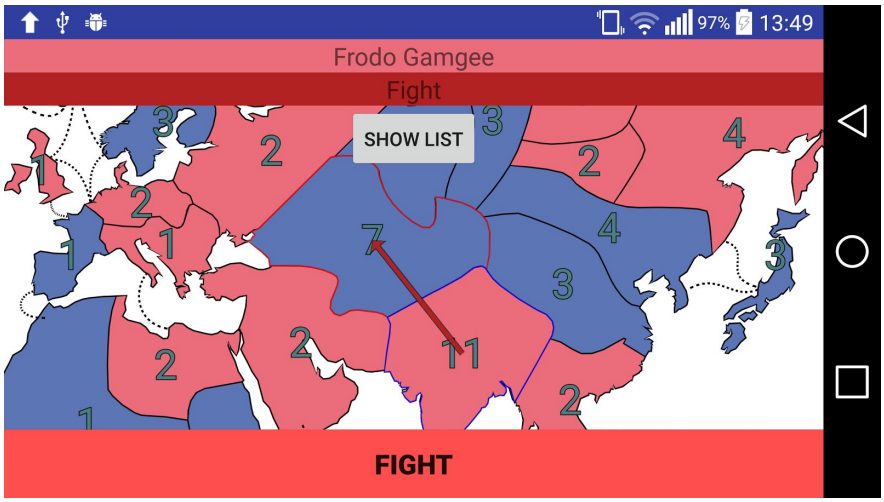
2.3.4 User interface
Interface is scalable to work with small screens, although small screens should not be a problem since most applicable Android devices (os version 4.0.3 or higher) have larger screens.

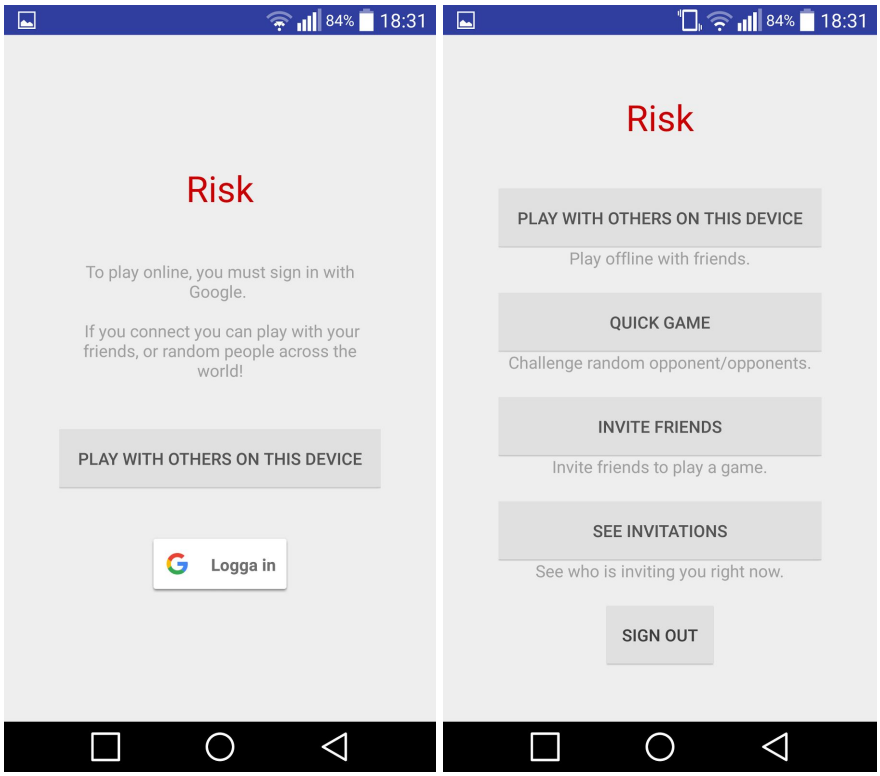
Gameboard with representation of territories, continents, armies etc. Zoomable, pannable.
“End turn”- button to end turn.
Menu button for various settings
Pressing on a country gives a list of options

2.4 References
Risk game: https://en.wikipedia.org/wiki/Risk_(game)

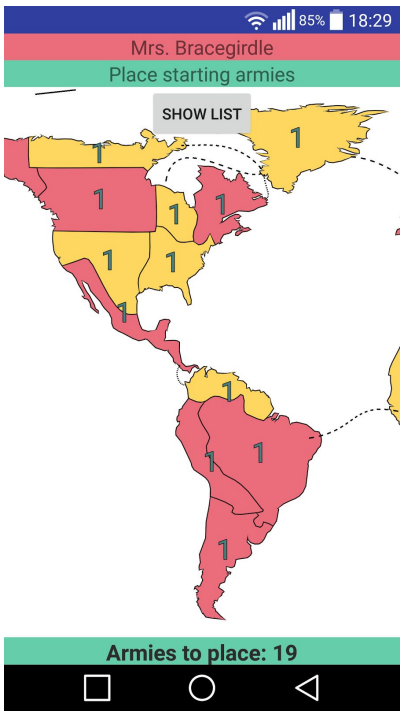
Appendix

Gui:

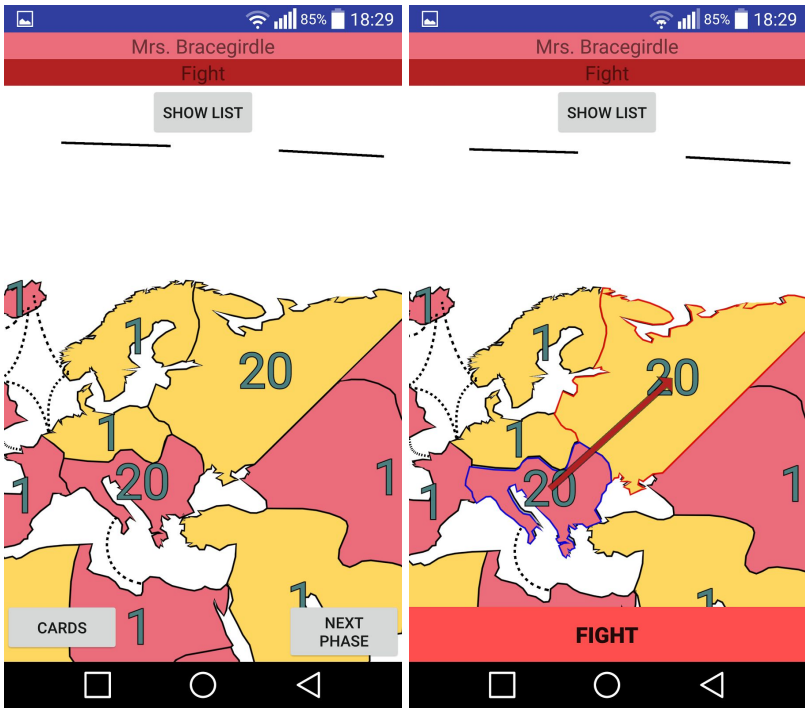




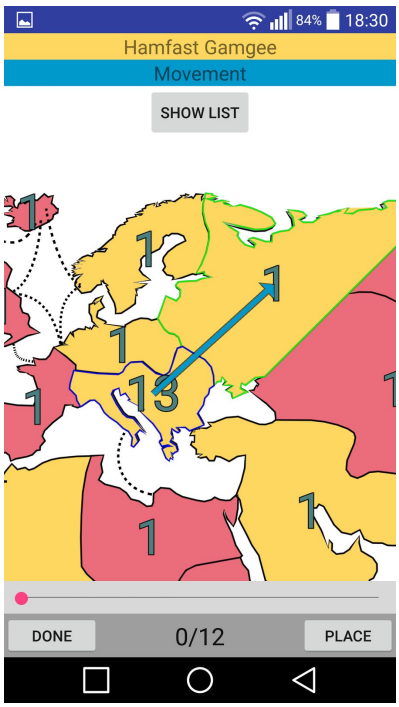
Start screens, not signed in (left), signed in (right)



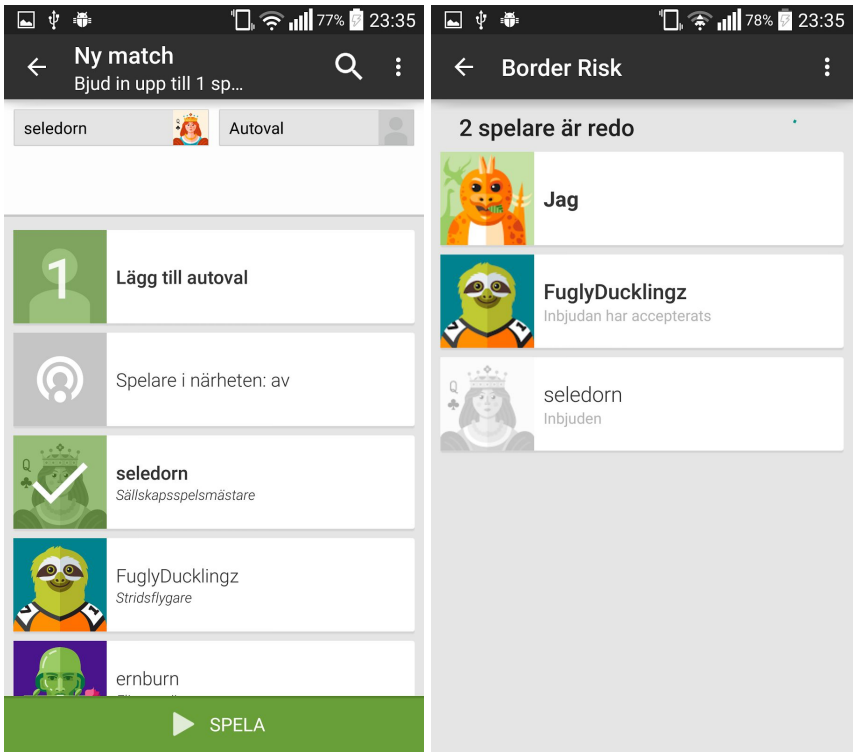
Place armies phase



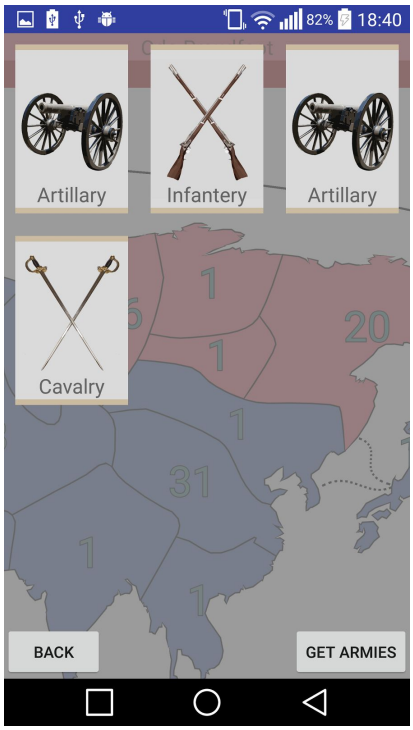
Fighting phase, before attacking and defending countries are set(left)
Fighting phase, after attacking and defending countries are set(right)



Moving game phase

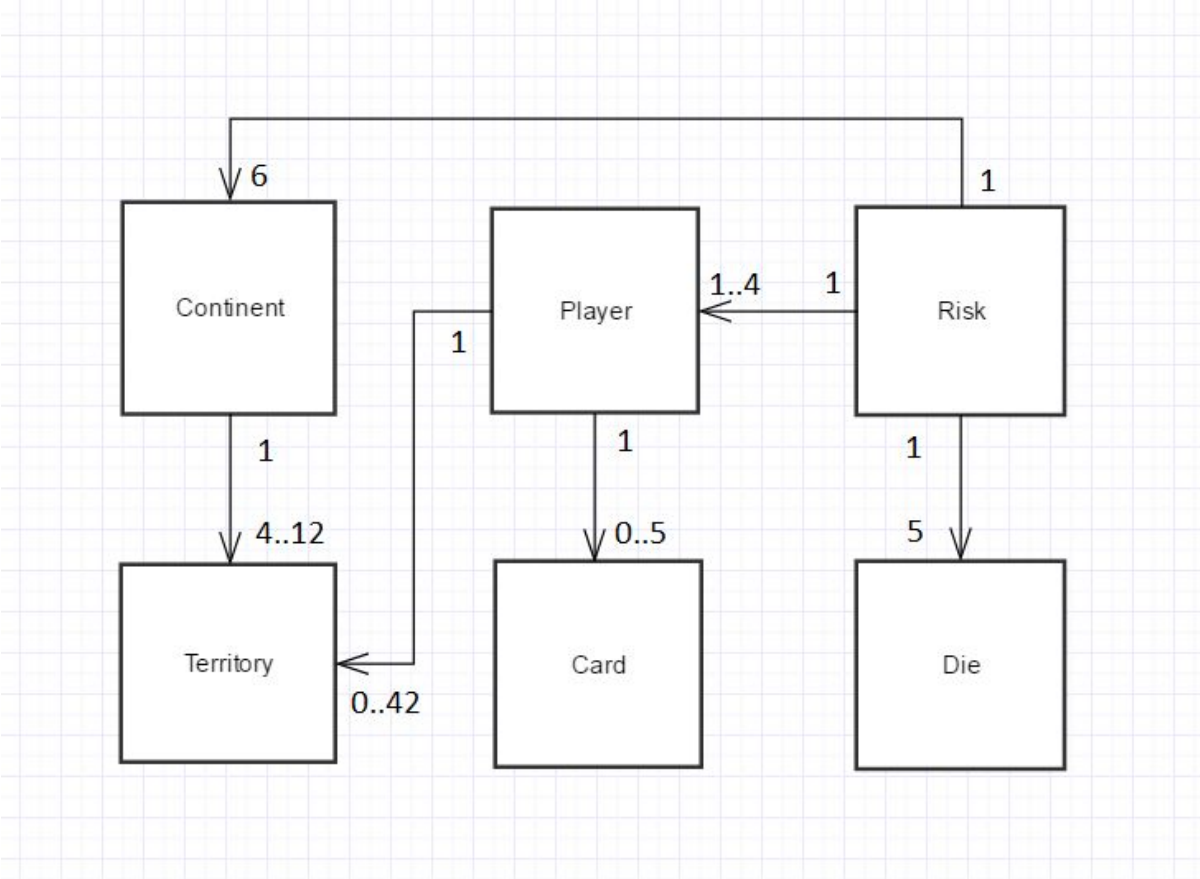


Sending invite (In users phone's language) (left)
Waiting room (In users phone's language) (right)



Card view

Domain model:



Use case model:
See external documents (git)

Use cases:
See external documents (git)