Requirements and Analysis Document for the Sankoss Project (RAD)

Contents:

Version: 1.0

Date: 19/3 - 2014

Authors: Fredrik Thune, Daniel Eineving, Mikael Malmqvist, Niklas Tegnander

This version overrides all previous versions.

1 Introduction

1.1 Purpose of application

The soul purpose of our project is to create an application that is engineered to entertain and substain our users in form of a light-weight computer game based on the board game Battle Ships. We aim to apply all the knowledge we've yet aquired during our first year at Chalmers into this project of our choice. For all the rules, definitions and further specific details about the game, se references!

1.2 General characteristics of application

Our application will be engineered for a desktop environment, (Linux/Windows/OS X) as a multi-player, based-turned game to be played over network. (or by yourself against an artificial intelligence)

The game will, as mentioned use a turn-based player alternation with no time constraint for each turn. This means that a player must choose to end its turn manually by shooting at a enemy grid.
The game will come to an end whenever a player has successfully neutrulized its opponents battle ships, by hitting the corresponding lenght of the ship, or when a player decides to cancel the current game.
To represent the game visually we'll use a semi-3D graphical user interface similar to the game board for the real board game.

1.3 Scope of application

At start-up the player will decide whether it wants to play against an artificial intelligence or a human opponent. (over network) We will NOT support several players on a single split screen. Therefore network connectivity is a requirment for all players.

1.4 Objectives and success criteria of the project

1. A player should be able to establish a connection to another player through a dedicated server

2. A player should be able to finish a game (see Definition) against a human opponent using a GUI

1.5 Definitions, acronyms and abbreviations

- Java: Platform independent programming language
- JVM: Java Virtual Machine
- GUI: Graphical user interface
- JRE: The Java Run time Environment. Additions software needed to run a Java application
- Host: Computer to run the game on (a player)
- Dedicated server: Server to route the player to the host
- Round: Successfully ended game with a winner and a loser
- Turn: In each turn a player chooses a target in the oppenent's grid to shoot at

2 Requirements

2.1 Functional requirements

The player(s) should be able to;

1. Start single player game.
2. Start multi player game.
    a. Create a room.
    b. Join specific room.
3. Start game within a room.
    a. Select nationality.
    b. Place ships and rotate ships on the grid.
4. Take a turn. During the turn, he or she can
    a. Aim at the enemy board, and fire at a target location.
    b. Send prerecorded taunts to the other player.
    c. Flag an enemy grid cell.
    d. End the turn.

5. Exit the application. Will end turn and round.

2.2 Non-functional requirements
Possible NA (not applicable).

2.2.1 Usability

The game mechanics should be so self explanatory that a normal user are able to play the game right away. Tests with two to four non-computer savvy people should make sure that the interface follows the criteria.

2.2.2 Reliability

Possible NA (not applicable).

2.2.3 Performance

A client-side action initiated by the player should not exceed 1 sec. response time, while the server is allowed up to 30 sec response time before the game shuts down because of slow internet connection.

2.2.4 Supportability

We are only going to add support for desktop. There should be easy to create an AI to play against the player.
It should be an easy task to switch the GUI to another - e.g from boats to space ships

2.2.5 Implementation

The application and java JRE needs to be installed on all hosts.

2.2.6 Packaging and installation

The application is packaged in a JAR-archive format with all dependencies included. It should be trivial to start this archive through the JVM.

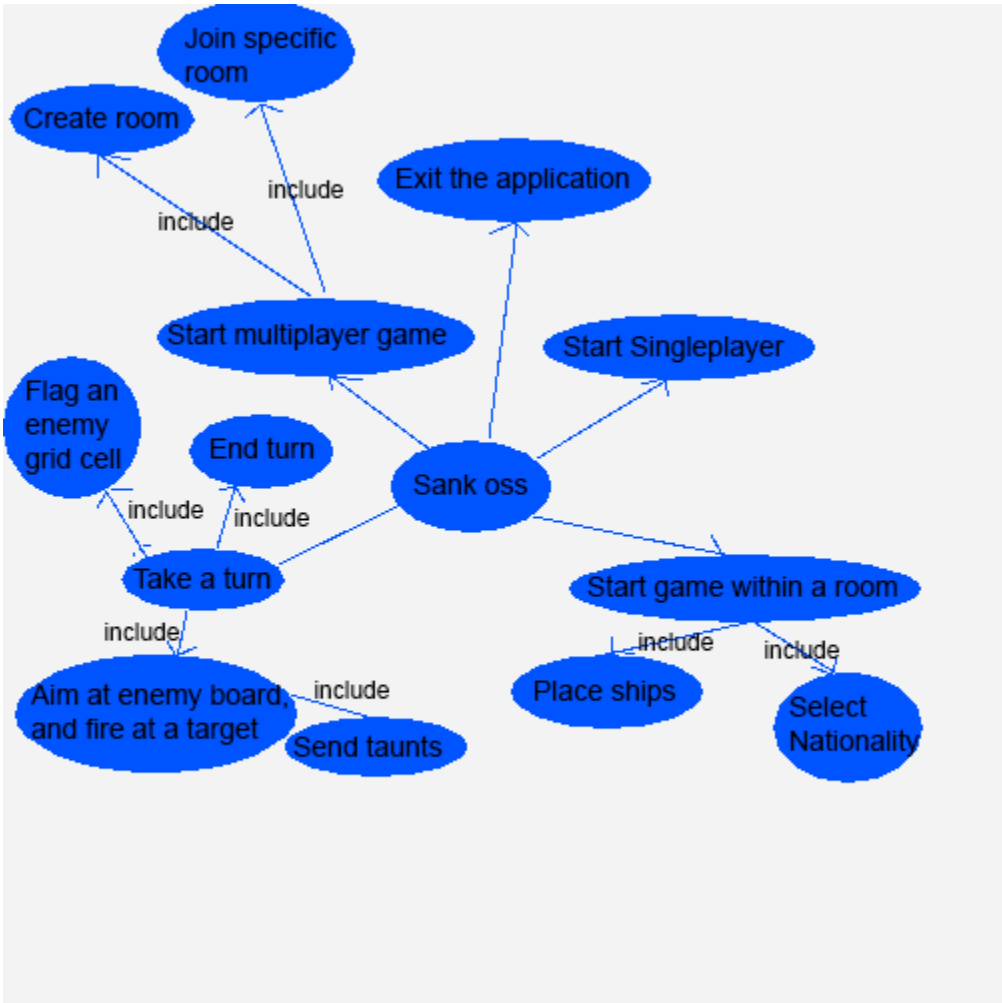
2.2.7 Legal

The license that is being used for the project is the MIT license. For URL, see References.

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. Join Room
2. Enter Game
3. Shoot at Enemy
4. Place Ships
5. Create Room
6. End Game
7. Singleplayer
8. Place Flags (Aimboard)
9. Set Nationality

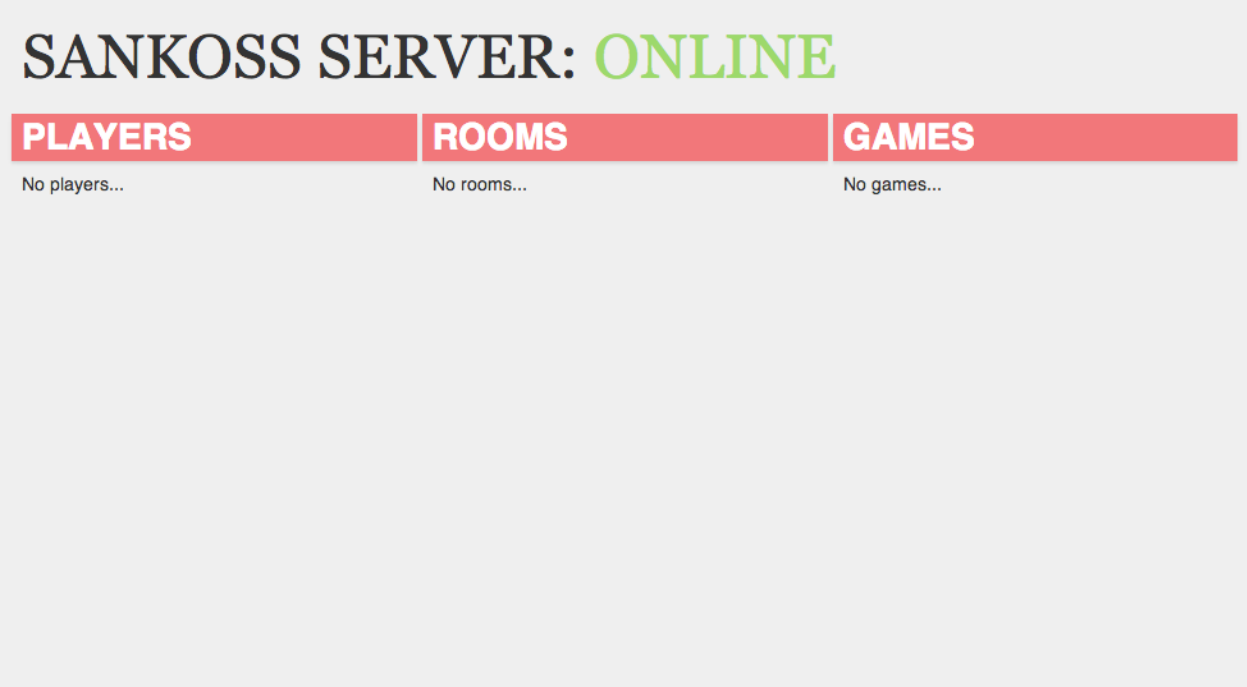
2.3.3 Domain model

See APPENDIX

2.3.4 User interface

Application will use a fixed GUI and must be able to adapt to different screen sizes and resolutions. See APPENDIX for screens and navigational paths.
Text to motivate a picture.

The server side of the application will have a web interface where a user can see a list of connected players, a list of current rooms and a list of current, ongoing games.



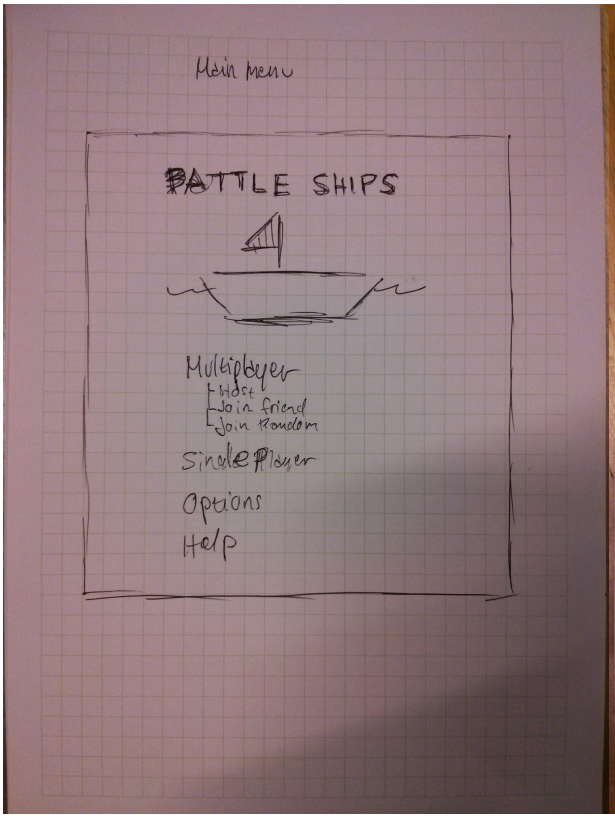
2.4 References

Battleships board game: http://en.wikipedia.org/wiki/Battleship_(game)
MIT: http://opensource.org/licenses/MIT

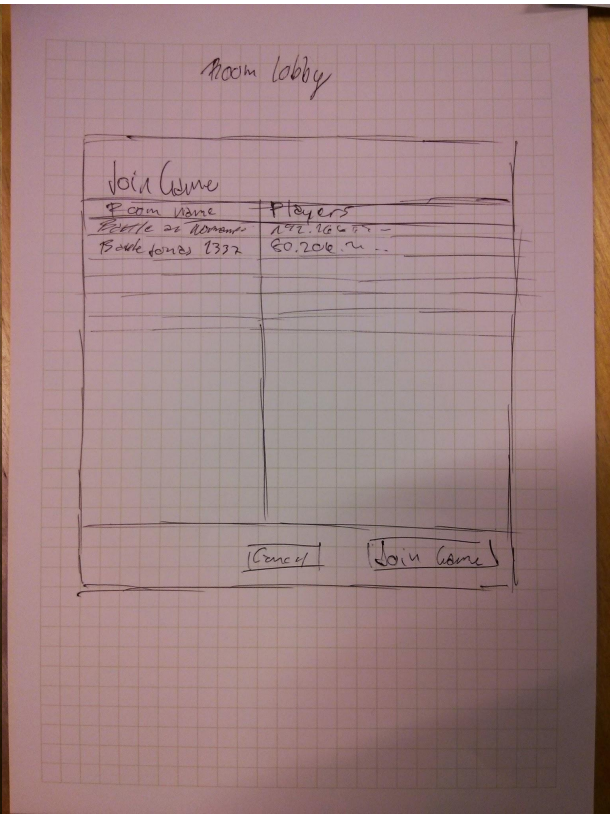
APPENDIX

GUI

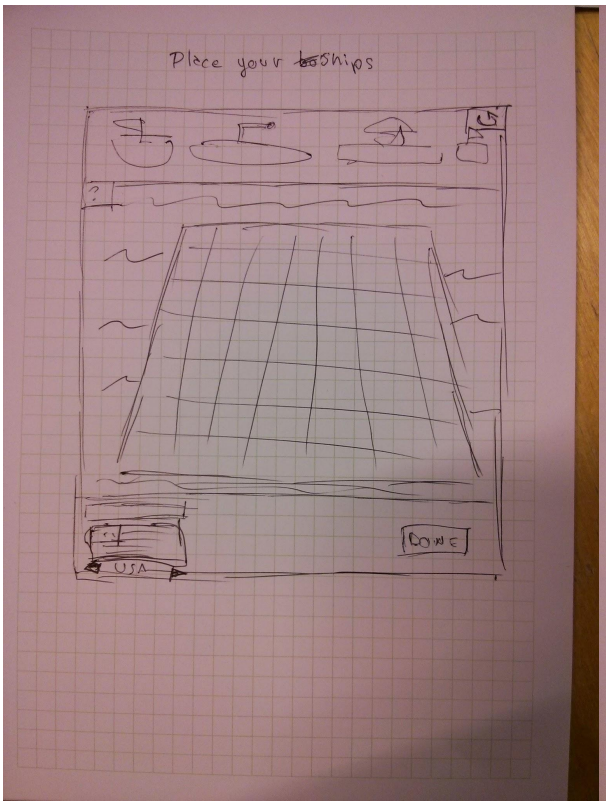
GUI mock-up



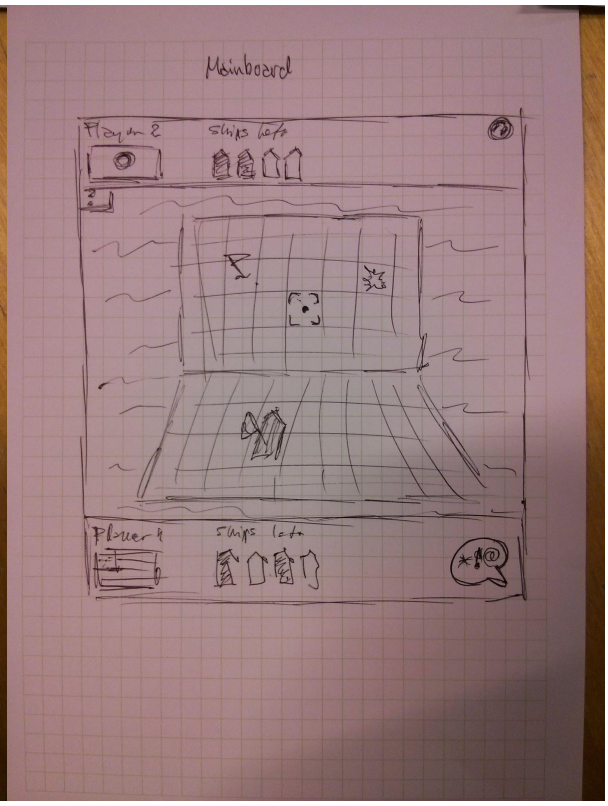
MainMenu



Lobby



Placement



InGame

Domain model

// TODO: insert analysmodell/uml/what ever

Use Cases

See attached Use Cases!