

# **Requirements and Analysis Document**

## **for Prjkté**

**Version: 2.0**

**Date: May 29, 2016**

**This version overrides all previous versions.**

Frej Karlsson

frejk@student.chalmers.se

Institutionen för Informationsteknik

Arvid Björklund

arvbjo@student.chalmers.se

Institutionen för Informationsteknik

Ludvig Ekman

eludvig@student.chalmers.se

Institutionen för Informationsteknik

Axel Bergrahm

cid@student.chalmers.se

Institutionen för Informationsteknik

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose of application . . . . .	1
1.2	General characteristics of application . . . . .	1
1.3	Scope of application . . . . .	1
1.4	Objectives and success criteria of the project . . . . .	1
1.5	Definitions, acronyms and abbreviations . . . . .	1
<b>2</b>	<b>Requirements</b>	<b>2</b>
2.1	Functional requirements . . . . .	2
2.2	Non-functional requirements . . . . .	2
2.2.1	Usability . . . . .	2
2.2.2	Reliability . . . . .	2
2.2.3	Performance . . . . .	2
2.2.4	Supportability . . . . .	3
2.2.5	Implementation . . . . .	3
2.2.6	Packaging and installation . . . . .	3
2.2.7	Legal . . . . .	3
2.3	Application models . . . . .	3
2.3.1	Use case model . . . . .	3
2.3.2	Use cases priority . . . . .	3
2.3.3	Domain model . . . . .	4
2.3.4	User interface . . . . .	4
<b>3</b>	<b>References</b>	<b>4</b>

## 1 Introduction

### 1.1 Purpose of application

This Projekt intends to create a fighting game which is a mix between Capcom's '*Street Fighter*' and Nintendo's '*Super Smash Bros*'. The game is planned to include several selectable maps and characters from which the player can choose from.

### 1.2 General characteristics of application

The application will be an offline multiplayer created for desktop (OSX, Windows and Linux). The application will have two player-controlled characters whom can move within an arena. Their role is to use and combine different attacks to inflict damage onto one another. When one of the characters is out of HP (*HitPoints*) they will lose and the game ends.

### 1.3 Scope of application

The application is a multiplayer game that needs atleast two players to play, i.e. you can not play the game alone. The users will not be able to save an ongoing game and no stats or similar will be saved. Every new game will be independent from previous games played.

### 1.4 Objectives and success criteria of the project

- It should be possible to run an instance of the game where we have a functional multiplayer, where the players goals is to eliminate the other players character.
- The application itself should be easily extendable, that one can with ease extend the selectables, for new characters or maps (for the developers).
- The game should be easy to understand and to play. This includes controls that are intuitive and easy to understand and also clear visual feedback for the players actions.

### 1.5 Definitions, acronyms and abbreviations

- **HP** – Health Points, the set of punishment a player can withstand before losing.
- **Stage** – An area where the players' quarrel is settled, the Arena.

- **Damage** – The amount of reduction of one players HealtPoints from the other players hit.

## 2 Requirements

### 2.1 Functional requirements

**The player should be able to:**

- Selcet a character
- Select a map
- Move around
- Attack
- Take damage
- Get eliminated

### 2.2 Non-functional requirements

#### 2.2.1 Usability

The program should be very intuitive and easy to understand, thus eliminating the need for excessive user manuals. Only a short manual with the controls will be available both ingame and when choosing map and character.

#### 2.2.2 Reliability

NA

#### 2.2.3 Performance

The program should not have any huge memory leakage. After a user does something it should not take more than 0.5 seconds before they receive some sort of feedback.

The program should not suffer from memory leaks.

## 2.2.4 Supportability

Initially the game is set to run on all standard desktop operating systems, i.e. Linux, OSX and Windows. The Game should however be easy to port to a mobile platform like Android or iOS and the code should be written so that only the GUI and some minor tweaks to the code should be necessary for it to run on those platforms.

Tests should exist for verifying all implemented use cases.

## 2.2.5 Implementation

The game will run using Java Environment (JRE) which needs to be installed on the host. The application need to be installed before running.

## 2.2.6 Packaging and installation

The program will be downloaded from git and run using the command 'gradle run'. Tests are run using 'gradle test'. A README.MD file will be available in the repository with instructions on how to play the game and how to run the program.

## 2.2.7 Legal

There are no legal issues.

## 2.3 Application models

### 2.3.1 Use case model

See Appendix for UML diagram and textual descriptions.

### 2.3.2 Use cases priority

#### High priority:

- Start application
- Fight
- Die
- Move
- End game

- Attack
- Take damage

**Normal priority:**

- Select character
- Fall off

**Low priority**

- Select map
- Special attack
- Knockback
- Open menu
- Close menu
- Change settings
- Pickup item
- Use item
- Drop item.

**2.3.3 Domain model**

See appendix.

**2.3.4 User interface**

The program uses a fixed GUI designed for a standard screen. See appendix for screenshots of the program and its GUI.

### 3 References

- Super Smash Bros game: [https://en.wikipedia.org/wiki/Super\\_Smash\\_Bros](https://en.wikipedia.org/wiki/Super_Smash_Bros)
- Street Fighter game: [https://en.wikipedia.org/wiki/Street\\_Fighter](https://en.wikipedia.org/wiki/Street_Fighter)

## Appendix

### GUI

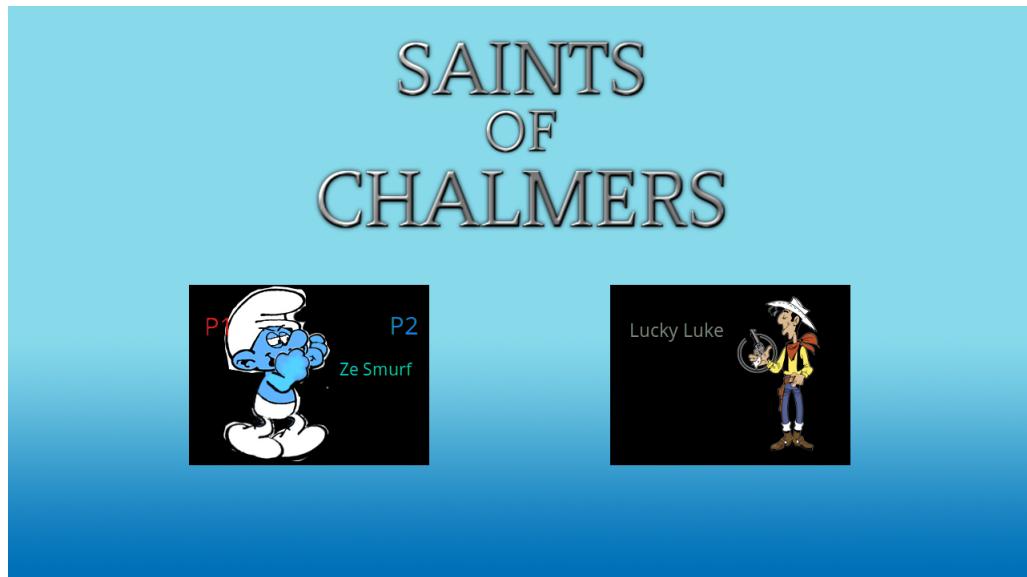


Figure 1: Start page, and choice of character. Player 1 and 2 is choosing the smurf character.

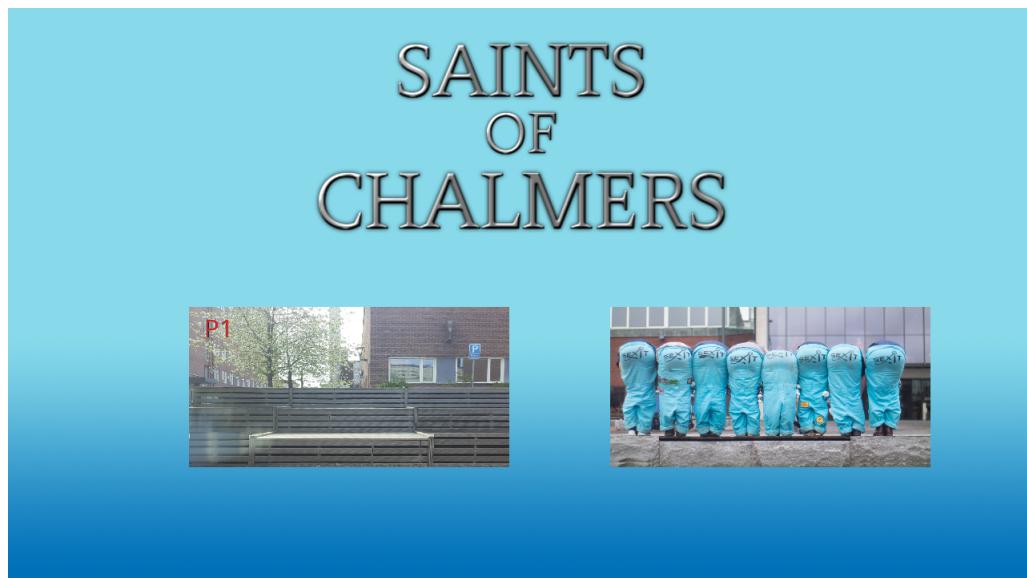


Figure 2: Choice of arena.

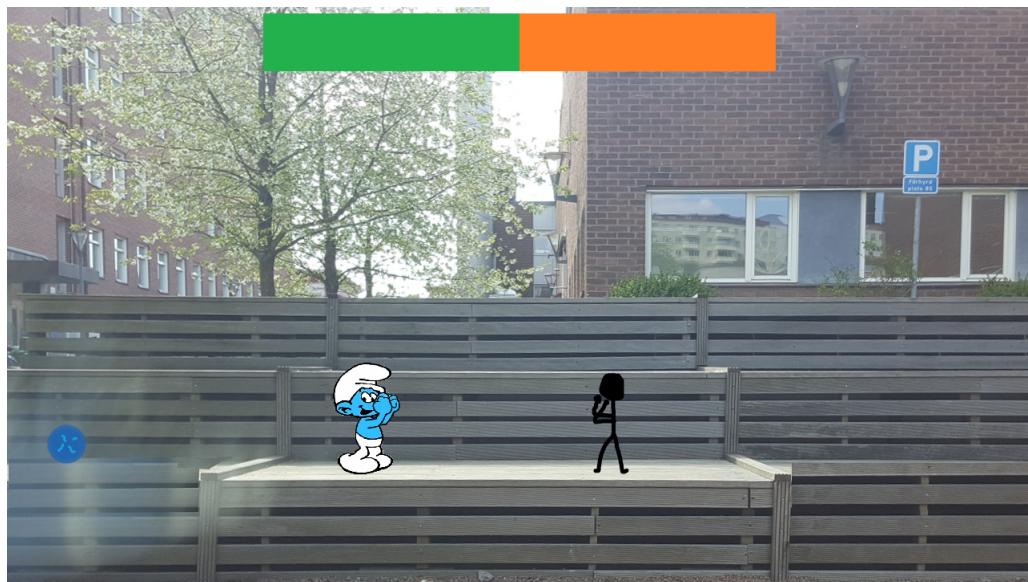


Figure 3: *The two characters are ready to fight in the arena.*



Figure 4: *Paus menu with three options: "Continue", "Character select" and "Exit".*

## Use case texts

See attached files in subfolder UseCases.

## Domain model

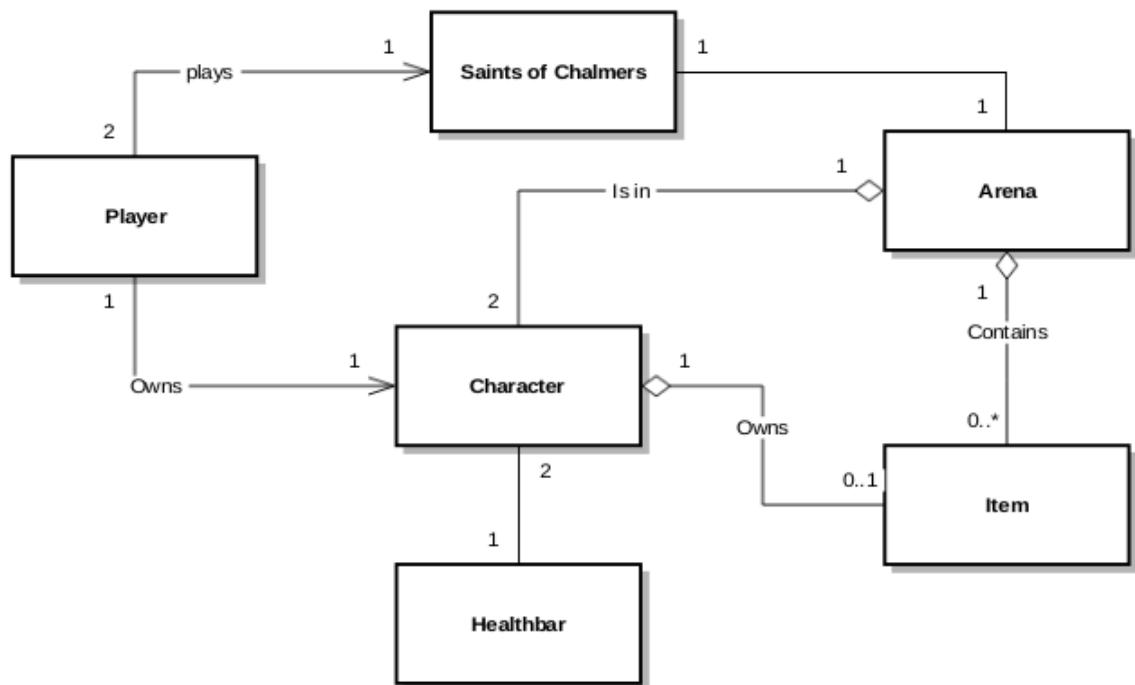


Figure 5: *Domain Model*.

## UML diagram

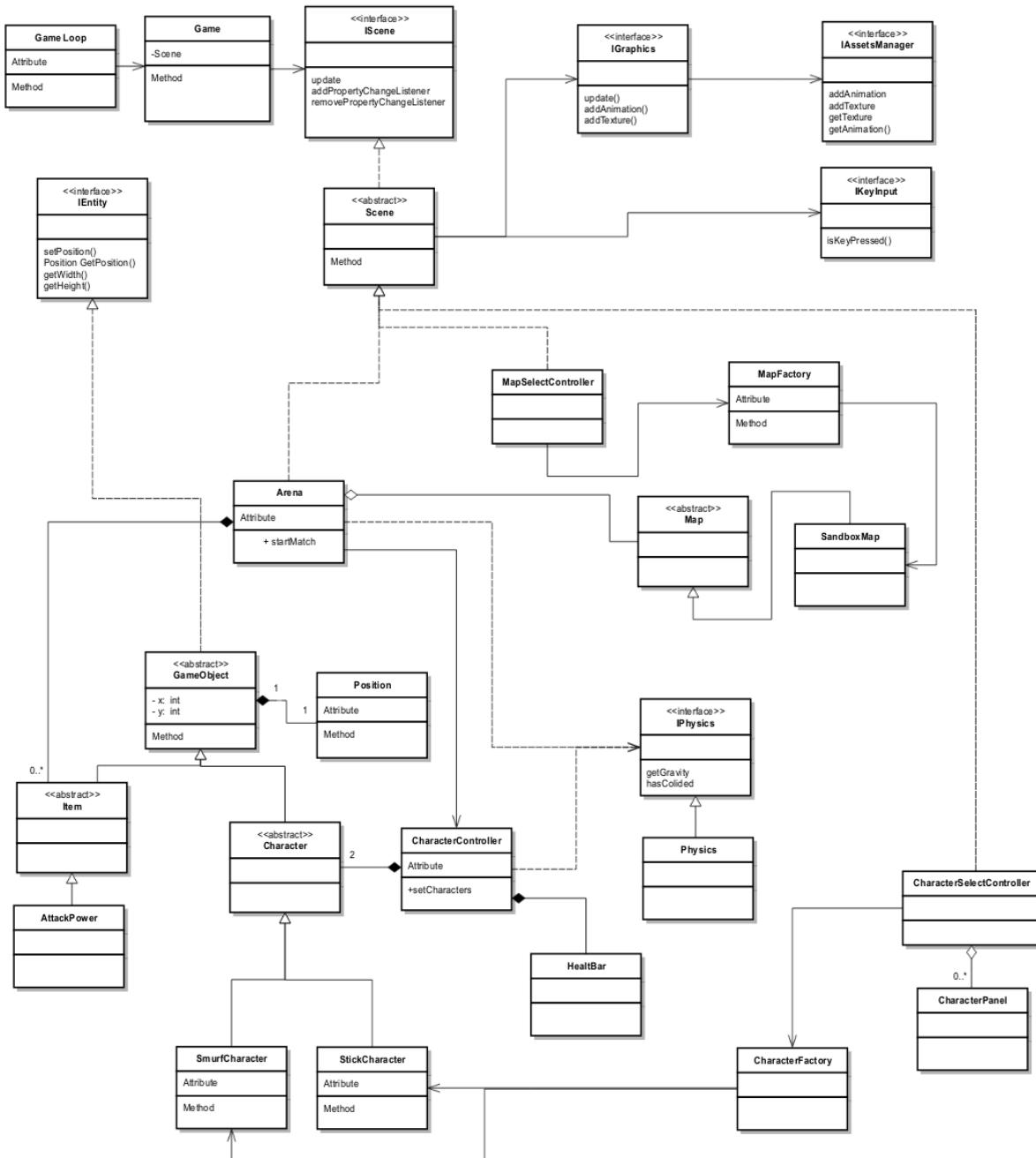


Figure 6: UML.