Requirements and Analysis Document for Rojarna.

Version: 1.0

Date 19/3-14

Author: Oskar Frost, Tobias Foughman Lind

This version overrides all previous versions.

# 1 Introduction

## 1.1 Purpose of application

The project aims to create a modern version of the classical Microsoft-game minesweeper. This version will include more features than the original, and enhance the user-experience.

## 1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), multiplayer application with a graphical user interface for the Winds/Mac/Linux platforms.

This adaptation of Minesweeper will include three different game-modes, “Campaign”, “Classic”, and “Free” mode. All which will include the classical gameplay, but with some twists.

The “Classic” mode will function exactly as the original.

The “Campaign” mode will guide the player through a set of stages, of all which will be needed to complete within a time-limit. Between and during games, the player have the option to buy power-ups to help him clear a stage. The power-ups will be purchased by a time-currency, gained by completing each stage with time left. The more the stages the player completes, the difficulty increases by increasing the grid and mines and decreasing the initial time for each stage.

## 1.3 Scope of application

(**Spara** framsteg i campanjen, spara bästa lösningstid (classic))

## 1.4 Objectives and success criteria of the project

1. It should be possible to play a full game of the classical minesweeper-game, randomly generated.
2. The campaign mode should be playable for at least one level, with all the power-ups functioning.

## 1.5 Definitions, acronyms and abbreviations

All definitions and terms regarding the core Minesweeper game are as defined in the reference section.

# 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).

## 2.2 Non-functional requirements

Possible NA (not applicable).

### 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

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

2.3.2 Use cases priority  
A list

### 2.3.3 Domain model

UML, possible some text.

### 2.3.4 User interface