

# Requirements and Analysis Document for

## ‘Tis but a Scratch!

### 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: 1.0

Date: 14-03-19

Author: Anna Nylander & Ivar Josefsson

This version overrides all previous versions.

# 1. Introduction

This section gives a brief overview of the project.

## 1.1 Purpose of application

The purpose of the game is to create an topdown 2D adventure game in wich two players can cooperate in completing puzzles and fighting enemies located in seperated areas.

## 1.2 General characteristics of application

The application will be a desktop, standalone and multi-player application with a graphical user interface for the Windows/Linux enviroment.

The application will be a realtime mutliplayer game over network in wich the player will be able move freely over the board, devided in a grid like the classic Pokemon och The Ledend of Zelda games. The player will also be able to fight enemies and interact with diffrent objects.

## 1.3 Scope of application

The application will not be able to save started games or collect any terminated games. The game will only use keyboard and therefore not support mouse or any other controllers. You will only be able to play mutliplayer on seperate PC's, you cannot play multiplayer on the same PC. The project will not implementent possibility for several players on the same computer.

## 1.4 Objectives and success criteria of the project

Multiplayer experience

Defeat enemies

Change rooms

Solve puzzles

## 1.5 Definitions, acronyms and abbreviations

User = Any user of the program

Main user = The player sitting at the host computer

## 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

Text to motivate a picture.

## 2.4 References

APPENDIX

GUI

Domain model

Use case texts