Project Descent

Software Requirements Specification

1.0

14/10/2015

**Table of Contents**

**REVISION HISTORY**

**DOCUMENT APPROVAL**

**1. INTRODUCTION**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 Overview

**2. GENERAL DESCRIPTION**

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

**3. SPECIFIC REQUIREMENTS**

3.1 External Interface Requirements

3.2 Functional Requirements

3.3 Use Cases

3.4 Classes / Objects

3.5 Inverse Requirements

3.6 Design Constraints

3.7 Logical Database Requirements

**4. ANALYSIS MODELS**

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

# **Introduction**

This document aims to explain the scope and specification of the board game Descent: Journeys In The Dark and what is require in order to create this game as a video game using Unity3D.

This document will cover the purpose for this product, scope, functions of the game along with any constraints and dependencies.

It will then go on to discuss what the game will look like such as the user interface along with any requirements the game may need which will be both functional and non-functional.

## **1.1 Purpose**

Like stated in the introduction, the purpose of this document aims ro explain the scope and specification of the board game which we will be creating in our chosen game engine.

The intended audience for this game both males and females of the age 14+ this is due to one of the core mehcnaics of the game being combat which may not be appropriate for anybody who is younger than this age.

## **1.2 Scope**

The aim of this project is to produce a game which will be similar in aspect to the board game counterpart. To do this we will be using 3 main programs which will be used will be Unity3D along with Monodevelop (Unitys intergrated development environment(IDE)) and Visual Studio which is another IDE however both are being used as it depends on the employees preferences.

The software will allow the player to play the game Descent as if it was a video game. This means it will adhere to all the same rulings as the board game which would include the following:

* Choosing a quest
* Choosing a hero for that quest
* Choosing starting equipment
* Being able to level up and choose new abilities
* Attack from both a melee and ranged distance
* Allow the rolling of dice (in this case random generated numbers(RNG))

However this does not mean that the player can do the following:

* Break any of the rules from the original board game such as line of sight. Any time these rules are broken through testing it will be seen as bug.
* Cannot choose starting equipment which does not belong to that players class.
* Cannot level up morew than once at a time.
* Attack from the wrong distance (such as a hero with a sword cannot attack a for which is not in an adjacent tile.
* Cannot roll dice which he does not need to roll.

This piece of software will have multiple goals and the first one will be that as stated before when the player is using this piece of software they will be able to play like they were playing the board game. The software aims to differ to the board game as little as possible and any change in gameplay will be because it would not be applicable to do so. (such as rolling a dice and the software knowing what number the user rolled.)

## **1.3 Definitions, Acronyms, and Abbreviations**

Operating System – OS

System Requirement Specification – SRS

## **1.**4 **Overview**

This document is split into five sections, section one is to outline the purpose of the software and inform readers of the SRS what each acronym and definitions mean. Section two is used to define the functions and contraints of the system. Section three is about the requirements set by the user and specific requirements for the system. Section four is showing what models were used to analyse the data. Section five is used to describe the ongoing process of keeping the SRS updated as the project moves forward.

# **2. General Description**

## **2.1 Product Perspective**

As mentioned above this product is only going to be a tiny section of the whole game ‘Descent’, this means that there is other parts of the system that will not be mentioned or made within this product. As we have decided to make the first quest in the game there is things that will have developed as the game goes on that cannot be part of the first quest.

As there is many other groups that have been set the same assignment, some of which will be based on the same quest but others may be different- but they will still be related to the same game.

Board games similar to ‘Descent’ include ‘Dugeons and Dragons’ which is another quest based game based loosly on a medival myth, there is also an overlord in this game, making it even more similar to ‘Descent’.

## **2.2 Product Functions**

## **2.3 User Characteristics**

The game should have a tile based movement system which allows the user all the actions they can do in the board game which are specified above. When a player cannot use a certain action the action button to do so will be greyed out and when clicked will display an error message saying why they cannot perform that action.

## **2.4 General Constraints**

One general constraint that may affect the users experience of this game is if they are running an older OS, as this game will not be tested across all available types / version of every OS it is possible that they could be unable to access part of the game if the OS hasn’t been tested on.

## **2.5 Assumptions and Dependencies**

No special assumptions or dependancies have been identified. It should be able to run on all operating systems.

# **3. Specific Requirements**

## **3.1External Interface Requirements**

### **3.1.1 User Interfaces**

All interaction with the user will be via an interface. Once the game has started the user will be prompted for an action, the game will then act upon that action and if the user does not respond within that time then the system will end their turn.

### **3.1.2 Hardware Interfaces**

A mouse, keyboard, at least one monitor along with a computer.

### **3.1.3 Software Interfaces**

An official copy of any Windows, MacOS.

The system shall be capable of running on any version of window, the system should make appropriate use of the OS. and the file management system to store and retrieve the game state.

## **3.3 Use Cases**

See separate files.

## **3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

The inverse requirements for this system are the following:

* The player must never be able to move more than the amount on their card unless under the effect of another card or ability.
* The player must never be able to win from below turn 3.
* The player must never be able to inflict more damage than is on their card unless under the effect of another card or ability.

## **3.7 Design Constraints**

One design constraint within this game is that all the language used in it will be in English so there will be a standard of the oxford English dictionary for all the text within the game. As the game is being produced in at the university there will a standard of ethics that will have to be observed when developing the game.

## **3.8 Logical Database Requirements**

We will be using text files to store the information about each hero, this will contain their stats and name. The player will choose from the list of characters and the system will access this file to get the base statistics.

# **4. Analysis Models**

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

## **4.1 Sequence Diagrams**

See separate file.

## **4.3 Data Flow Diagrams (DFD)**

See separate file.