SRS

(Software Requirements Specification)

Descent Journeys in the Dark: Second Edition

1.0 – 18/10/2016

Team Ascension

Jai Powles – Project Lead - 1519938

Tony Fraser – Lead Designer - 1530845

Alex Hunt – Lead Programmer - 1536026

Michael Sharpington - Lead Tester – 1407802

**Table of Contents**

Revision History 4

1.0 Introduction 4

Purpose 4

Document Conventions 4

Intended Audience and Reading Suggestions 4

Product Scope 4

References 5

2.0 Overall Description 5

Product Perspective 5

Product Functions 5

User Classes and Characteristics 6

Operating Environment 6

Design and Implementation Constraints 6

User Documentation 6

Assumptions and Dependencies 7

3.0 External Interface Requirements 7

User Interfaces 7

Hardware Interfaces 7

Software Interfaces 8

Communications Interfaces 8

System Features 8

System Feature 1 8

System Feature 2 (and so on) 10

4.0 Other Non-functional Requirements 10

Performance Requirements 10

Safety Requirements 10

Security Requirements 10

Software Quality Attributes 10

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
|  |  |  |  |

# 1.0 Introduction

## Purpose

The purpose of this document is for the videogame my group is creating, taking an established board game, “Descent: Journeys in the Dark” and putting it into an online game, with mechanics designed around how a board game will function. This game was designed as part of a university project, featuring a group of four.

## Document Conventions

When designing this piece of Documentation, certain styles were chosen to empathize the Severity and Priority of certain systems, when talking about severity, Highlights have been assigned to each system, Green for low, Yellow for Medium, Red for Critical. When it comes to priority, we simply place a number e.g. (3) next to the name of the system, between 1 and 5, 1 being High priority and 5 being backburner/ when other main tasks are complete. We shall also at least put a small detailed report on each system, saying what each members focus will be on that task.

## 

## Intended Audience and Reading Suggestions

This document is designed so that anyone has the ability to understand our goals, and how we go about achieving them. When reading through the document however, some sections will be more useful than others. For developers, the entire document should at least be helpful in our progression of the task, however sections 2 and 3 will definitely be the focus. For the marketing side of the project, section 4 will be a bigger focus, as it has the business rules of which they will have to abide to when designing advertisements. Testers will have section 3 as their reference point, as it goes into details on systems they will have to be testing. Documentation writers will have section 2 as their reference point, as it also has examples of our documentations style and actual physical reference.

## Product Scope

The software we are developing is the video game version of Descent Journeys In The

Dark. The benefits of this project for everyone is, learning how to work in teams and also getting used to using organisational software such as GitHub and Trello. The goal is to complete the game so that it meets the requirements set by our module tutor.

## References

<http://descent2e.wikia.com/wiki/Descent:_Journeys_in_the_Dark_(Second_Edition)_Wiki>

# 2.0 Overall Description

## Product Perspective

The product is both a follow-on member and also a new self contained product. The reason it is a follow-on member of a product family is because it’s a replica of the board game. The reason it is a new self contained product is because it’s the first product in its series to make it to a video game. However the creation of this product means that its created a larger system itself. There are many expansion packs for the descent board game which can be developed and implemented into the video game. This means that the original version of the board game must be able to withstand multiple version updates, so expansion packs can be implemented.

## Product Functions

The major functions the product must perform are:

* At least one functioning quest
* The turn based mechanic of the game
* Heroes and the overlord must be able to perform all their relevant actions

## User Classes and Characteristics

Developer user – This user will be part of the development team, they will use the product to implement new updates and fix bugs. This user will have access to all the files regarding the software.

Tester User – This user will test the software and find bugs. They will then report this back to the development users. This user will only have access to the build.

Customer User – This user will use the software as a for of entertainment. They are at the end of the chain in terms of users and will only have access to the build.

## Operating Environment

The software is going to run on Windows OS. The software is going to made in Unity, currently version 5.4.1f.

We are also using the latest version of the SQL database browser and the relevant plugins for this software. If the SQL plugins do not work, the game will break.

## 

## Design and Implementation Constraints

Some constraints include:

* We only have 12 weeks to develop the software
* Having to use the same unity version throughout the assignment
* The game has to be developed to cater to everyone’s skill level

## User Documentation

User documentation can be found online at the Descent wiki page. The user manuals and quest guide cannot be duplicated for copyright reasons.

## Assumptions and Dependencies

* User has Windows OS
* Everyone has access to a PC that can run unity
* For developing – everyone has an internet connection to use GitHub and trello
* For developing – everyone understands OO programming and is familiar with the unity interface.

# 3.0 External Interface Requirements

## User Interface

The main interface the player will be interact with is the main menu. It will showcase

The new game button, load game button and exit game button. When the player clicks

“New Game”, the next menu will be a group of drop down menus. These will showcase

all of the appropriate characters that the player can choose. When the player has chosen

Their character, the game will receive information from the database and relay this back

to the UI and display the stats for each character. On the main game UI, for the player

to complete their turn, they must press the Enter key, and the appropriate character’s

sprite on the side will be highlighted. In the bottom right corner is the dice, to roll the

dice they must press the D key, and the dice script will choose a random number with

set outcomes. There is also the two drop down menus that allow for equipping items,

and the other for attack.

## Hardware Interfaces

This game is designed for Windows PC, and requires a keyboard and mouse.

## Software Interfaces

The game has a specific connection to the DB Browser, where our database stays. The

game has a script for the character selection that takes all of the character data from the

database and converts it to a string and displays it. This also runs in the same way in

the main in-game UI, except it only takes the health and fatigue points data.

## System Features

## Turn Based System

Description and Priority

This is the basic form of gameplay that we have to implement. This is high priority because our product is based around this type of gameplay.

## Character Selection

This is the way each player plays as a different character. This is high priority because

there needs to be 4 different variations of character in the game.

## Menu Navigation

This is how the players will be able to enter the game and exit it too. This is high priority

# because it is a basic necessity for the player to exit the game at any point.

# 4.0 Other Non-functional Requirements

## Performance Requirements

There are no other performance requirements other than the compulsory requirements set by the assignments. However if the software is completed a while before the deadline, then time may be allocated to improving the graphics of the game.

## Safety Requirements

N/A

## Security Requirements

There are not many safety requirements for our software. This is because our software will not include any online features so you do not need an internet connection to play. There are also no micro-transactions, so the game will not be holding any bank account details from customers. The only real safety issue is making sure the game doesn’t have any memory leaks, to prevent the software crashing.

## Software Quality Attributes

The software needs to have a high level of adaptability, flexibility, maintainability, testability and portability. This is because the game could receive regular updates to implement the expansions packs. The software therefor needs to be robust enough to deal with these regular updates and implement the changes with as little bugs as possible.

The game needs to posses a certain amount of portability if we, in the future, decided to develop the software for Mac or mobile.