***DungeonOfDooom***: Requirements Document (version 1.0)

Project: DungeonofDooom

Due Date: 16.12.2016

1. Introduction

This document contains the system requirements for *DungeonofDooom*. These requirements have been derived from Moodle.

1.1 Purpose of This Document

This document is intended to guide development of *DungeonofDooom*. It will go through several stages during the course of the project:

1. **Draft:** The first version, or draft version, is compiled after requirements have been discovered, recorded, classified, and prioritized.
2. **Proposed:** The draft document is then proposed as a potential requirements specification for the project. The proposed document should be reviewed by several parties, who may comment on any requirements and any priorities, either to agree, to disagree, or to identify missing requirements. Readers include lecturers. The document may be amended and reproposed several times before moving to the next stage.
3. **Validated:** Once the readers have agreed to the requirements in the document, it is considered validated.
4. **Approved:** The validated document is accepted by representatives of each party of reader as an appropriate statement of requirements for the project. The developers then use the requirements document as a guide to implementation and to check the progress of the project as it develops.

1.2 How to Use This Document

We expect that this document will be used by people with different skill sets. This section explains which parts of this document should be reviewed by various types of readers.

**Types of Reader**

Our readers are Senior Lecturer Julian Padget and Charlie Ann Page.

**Technical Background Required**

To understand code’s structure and development phases, readers should have knowledge on computer science.

1.3 Scope of the Product

To analyse, specify, design and implement a multi-player on-line game.

1.4 Overview of the Requirements Document

For the requirements a team was required. Test-driven development should be applied. Scrum meetings should be made every week. Agile development phases should be applied. For coding, application of MVC should be used. 3-tier architecture should be made. Mixed languages should be used for both client and server side.

2. General Description

This section will give the reader an overview of the project, including why it was conceived, what it will do when complete, and the types of people we expect will use it. We also list constraints that were faced during development and assumptions we made about how we would proceed.

There are functional requirements for project development.

Human players should select his/her component (bot or multiplayer). Map of the dungeon should be created with walls, passages and coins. Coins are the scores of the game. A dungeon can be

of arbitrary size. A dungeon should contain at least as much gold as is required to win, and at least one exit passage. Player’s or bot’s location(s) should be shown on the map visualization. Players should move with keys, pick up gold and looking to find about the current room and exit.

2.1 Product Team Perspective

Our team consists of 6 people.

2.2 Product Functions

Login and registration function should be applied. A database should be created for login and registration values. Player can move by key press functions in the dungeon. In the menu screen, users can select game type and levels, components with buttons. Score should be calculated by the coinscollected function and score should be shown in the score table section. Player’s location should be called from server and needed to be updated.

2.3 User Characteristics

According to customer meeting notes; target is everyone. It is an online game and it is not requiring any technical background, training or education.

2.4 General Constraints

Customer didn’t specify any specific languages. So in the sprint, team decided to use Java Code . For tile creation and parsing JSON is used. For interface applications html and JavaScript are applied. Eclipse platform is used. MVC is made with model, controller and view sections.

2.5 Assumptions and Dependencies

Finished product can be delivered over the internet. It requires administrator only for adding a new player. No need for specific skills to play the game. Project should be finished until 16 December 2016.

2.6 Project Budget

There is no specific budget according to customer meetings.

3. Specific Requirements

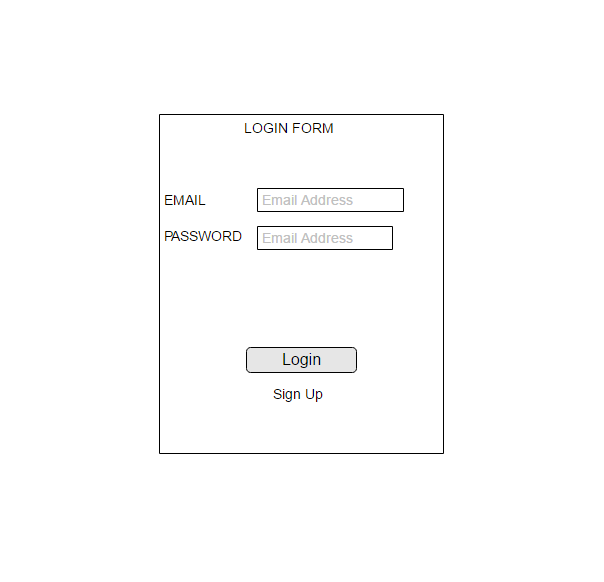
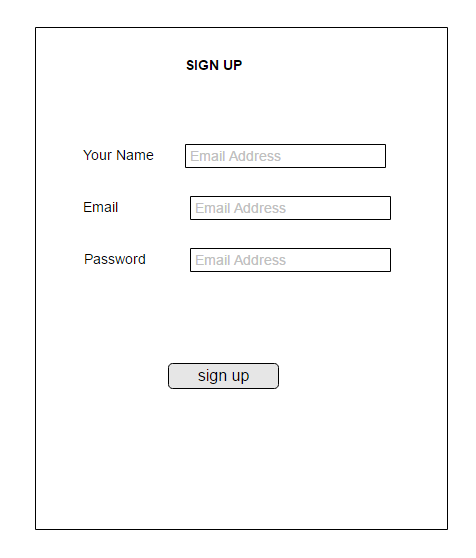
This section of the document lists specific requirements for *DungeonOfDooom*. Requirements are divided into the following sections:

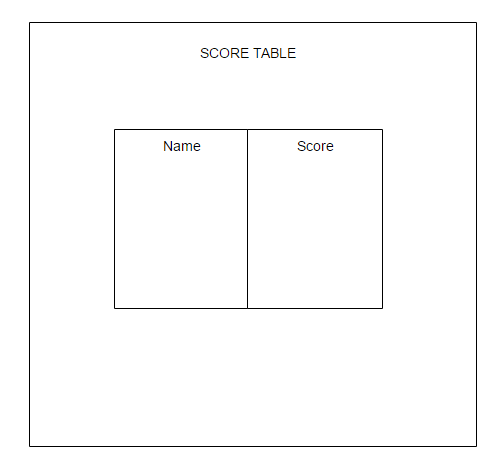
1. User requirements. These are requirements written from the point of view of end users, usually expressed in narrative form.
2. System requirements. These are detailed specifications describing the functions the system must be capable of doing.
3. Interface requirements. These are requirements about the user interface, which may be expressed as a list, as a narrative, or as images of screen mock-ups.

3.1 User Requirements

Every week sprints should be made. For customer, Trello may be good choice for tracking development. Github accounts should be created for each team member for code and document sharing. Timescale produce should be created for each development phase. Object oriented design should be applied. CRC cards and use cases should be written. Customer left the decision to the team for multiplayer screen. Turn based or concurrent selection will be discussed. Based on levels, algorithms will be improved. Score can be calculated by collecting all the coins in the rooms. If the player collects every gold, (s)he can exit and view the score. At the end project team and individual reports should be submitted. And team and individual questionnaires should be written.

3.2 System Requirements  
*//List detailed system requirements here. If your system is large, you may wish to break this into several subsections.*

**3.3 Interface Requirements  
MOCK UPS

******