

Group 11 SRS Document

February 8, 2019

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Definitions, Acronyms, and Abbreviations	1
1.4	References	2
1.5	Overview	2
2	Overall Description	2
2.1	Product Perspective	2
2.2	Product Functions	2
2.3	User Characteristics	2
2.4	Constraints	2
2.5	Assumptions and Dependencies	2
2.6	Apportioning of Requirements	2
3	Functional Requirements	2
4	Non-Functional Requirements	6
4.1	Look and Feel Requirements	6
4.1.1	Appearance Requirements	6
4.1.2	Style Requirements	6
4.2	Usability and Humanity Requirements	6
4.2.1	Ease of Use Requirements	6
4.2.2	Personalization and Internationalization Requirements	6
4.2.3	Learning Requirements	6
4.2.4	Understandability and Politeness Requirements	6
4.2.5	Accessibility Requirements	6
4.3	Performance Requirements	6
4.3.1	Speed and Latency Requirements	6
4.3.2	Safety-Critical Requirements	6
4.3.3	Precision or Accuracy Requirements	6
4.3.4	Reliability and Availability Requirements	6
4.3.5	Robustness or Fault-Tolerance Requirements	6
4.3.6	Capacity Requirements	6
4.3.7	Scalability or Extensibility Requirements	6
4.3.8	Longevity Requirements	6

4.4	Operational and Environment Requirements	6
4.4.1	Expected Physical Environment	6
4.4.2	Requirements for Interfacing with Adjacent Systems . .	6
4.4.3	Productization Requirements	6
4.4.4	Release Requirements	6
4.5	Maintainability and Support Requirements	6
4.5.1	Maintainance Requirements	6
4.5.2	Supportability Requirements	6
4.5.3	Adaptability Requirements	6
4.6	Security Requirements	6
4.6.1	Access Requirements	6
4.6.2	Integrity Requirements	6
4.6.3	Privacy Requirements	6
4.6.4	Audit Requirements	6
4.6.5	Immunity Requirements	6
4.7	Cultural and Political Requirements	6
4.7.1	Cultural Requirements	6
4.7.2	Political Requirements	6
4.8	Legal Requirements	6
4.8.1	Compliance Requirements	6
4.8.2	Standards Requirements	6

List of Tables

List of Figures

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

Player - The central playing piece of the game and the avatar that the user controls to interact with the game

Map - A network of interconnected rooms that form the available playing space for the *player*

Levels - A series of stages of incrementing difficulty that the *player* traverses

Spawn Room - The starting room that the *player* spawns in when entering a map for the first time

Inventory - The space that allows a *player* to store items, funds, and equipment

Skills - Abilities that a *player* can acquire from playing the game that improve the combat effectiveness of the *player*

Entities - Objects that the *player* can interact with

Crafting - The act of combining two or more items to create a new item

Ingredients - The material items that are consumed during crafting

Movement Key - A user input that corresponds to moving the *Player* in a cardinal direction (i.e. *North Movement Key* corresponds to moving the *Player* piece in the northern cardinal direction)

Basic Attack Key - A user input that corresponds to the *Player* performing a *Basic Attack* in a cardinal direction (i.e. *North Movement Key* corresponds to moving the *Player* piece in the northern cardinal direction)

Special Attack Key - A user input that corresponds to the *Player* using a

Basic Attack - The primary combat method of the *Player*

Enemy - Hostile game pieces that will attack the *Player*

Boss - A specialized *Enemy* acting as the final challenge of a *Map*

Agrovate - To make an *Enemy* aware of the *Player's* presence

Special Attack - Attacks obtained with *Skills* that have special effects

1.4 References

1.5 Overview

2 Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements

3 Functional Requirements

VP1. User

BE1.1 The user wants to start game

- i. The system presents a menu with the option of starting a new game or loading a previous save file
- ii. The user must be able to select either option of starting a new game or loading a previous save
- iii. If load game is selected, a list of all save files is presented
- iv. Each save file must be selectable

BE1.2 The user starts a new game

- i. A *Level 1 Map* must be generated
- ii. The *player* must be placed in the *Spawn Room*
- iii. The *player's Inventory* must be created

BE1.3 The user loads a game

- i. The map must be loaded from the save file
- ii. The *player* must be placed in the *Spawn Room*
- iii. The *player's Inventory* must be loaded from the save file
- iv. All previously attained *Skills* must be loaded from the save file

BE1.4 The user wants to traverse the map

- i. The *player* must move in the direction given by the user's input
- ii. Each room must have at least one passage to another room or level
- iii. The *player* must be able to enter through each passage
- iv. Each room must have associated *entities*
- v. The state of each room must be saved when exited
- vi. A newly entered room must generate all of its *entities*
- vii. A previously entered room must reload its most recently saved state

BE1.5 The user wants to buy an item

- i. Each Map generated must have at least one *Shop*
- ii. Each Shop must have a list of items with an associated cost
- iii. The *Player* must be able to interact with the shop
- iv. The *Player* must be able to purchase items from the Shop if they have sufficient funds in their *Inventory*

BE1.6 The user wants to craft an item

- i. The user must be able to open a menu listing all of the player owned items
- ii. The *Player* must be able to combining two or more items in their *Inventory* to create new items

- iii. If an item is *Crafted*, it is added to the inventory and its *Ingredients* are removed from the Inventory
- BE1.7 The user wants to move the *Player*
 - i. The user inputs a *Movement Key* without prompt from the system
 - ii. The *Player* responds by moving in the direction of the inputted *Movement Key* (i.e. *North Movement Key* corresponds to moving the *Player* piece in the northern cardinal direction)
 - iii. The *Player* will not be able to move through collidable *Entities*
- BE1.8 The user wants to perform a *Basic Attack*
 - i. The user inputs a *Basic Attack Key* without prompt from the system
 - ii. The *Player* responds by performing a *Basic Attack* in the direction of the inputted *Movement Key* (i.e. *North Movement Key* corresponds to moving the *Player* piece in the northern cardinal direction)
 - iii. *Basic Attacks* that hit collidable *Entities* that are not *Enemies* will be destroyed on those *Entities*
- BE1.9 The user initiates combat with an *Enemy*
 - i. The *Player* *Agrovates* an *Enemy* by attacking it
 - ii. *Basic Attacks* that hit an *Enemy* will apply damage to that *Enemy*
 - iii. *Special Attacks* that hit an *Enemy* will apply damage and other effects defined by the attack
- BE1.10 The user initiates combat with a *Boss*
 - i. The *Player* *Agrovates* a *Boss* by attacking it
 - ii. *Basic Attacks* that hit a *Enemies* will apply damage to those *Enemies*
 - iii. *Special Attacks* that hit *Enemies* will apply damage and other effects defined by the attack
- BE1.11 The user enters a level transition

- i. Requirement
- ii. Requirement
- iii. ...

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

4.1.2 Style Requirements

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

4.2.2 Personalization and Internationalization Requirements

4.2.3 Learning Requirements

4.2.4 Understandability and Politeness Requirements

4.2.5 Accessibility Requirements

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

4.3.2 Safety-Critical Requirements

4.3.3 Precision or Accuracy Requirements

4.3.4 Reliability and Availability Requirements

4.3.5 Robustness or Fault-Tolerance Requirements

4.3.6 Capacity Requirements

4.3.7 Scalability or Extensibility Requirements

4.3.8 Longevity Requirements

4.4 Operational and Environment Requirements

4.4.1 Expected Physical Environment

4.4.2 Requirements for Interfacing with Adjacent Systems

4.4.3 Productization Requirements

4.4.4 Release Requirements

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

4.5.2 Supportability Requirements

4.5.3 Adaptability Requirements

4.6 Security Requirements

4.6.1 Access Requirements

4.6.2 Integrity Requirements